

Proceedings of the Workshop of Undergraduate Work

XXI Brazilian Symposium on Computer Graphics and Image Processing

October 12-15, 2008, Campo Grande, MS, Brazil

Edited by

Soraia Raupp Musse (PUCRS)
José Pinheiro Queiroz-Neto (CEFET/AM)

Índice

Prefácio	iv
Comitê de Programa	v
Revisores	vi
Extração de Informações de Cor e Forma para Detecção de Placas de Trânsito em Imagens	1
<i>Ricardo Cezar B. Rodrigues, Sergio Roberto M. Pellegrino e Carlos Henrique Q. Forster</i>	
Equalização de Histograma para Melhoria de Contraste de Imagens Digitais Codificadas pela Técnica de Quantização Vetorial	5
<i>Marcelo Portela Sousa, Marcelo S. Alencar, Carlos Antônio Vieira Vasconcelos Júnior e Waslon Terllizie A. Lopes</i>	
Um Novo Algoritmo para Detecção de Contorno de Melanomas em Imagens de Pele	9
<i>Marilú G. N. M. da Silva e Ana Lúcia B. Candeias</i>	
Método de Classificação de Pontos em Esqueletos de Resolução Aumentada	13
<i>Diógenes Corrêa Canoza e André Vital Saúde</i>	
SketchSim: Interactive Simulation of Ink Brushstrokes	17
<i>Nicole Sultanum e Silvio B. Melo</i>	
Uso de Saliências do Contorno via Esqueletização para Caracterização de Formas	21
<i>Glauco V. Pedrosa, Marcos A. Batista, Henrique C. Fernandes e Celia A. Z. Barcelos</i>	
Técnicas Simples para Visualização Foto-realista de Terrenos em Tempo Real Usando Placas Gráficas Programáveis	25
<i>Ricardo Gomes Leal Costa e Waldemar Celes</i>	
Classificação Automática de Imagens do Acervo de Obras de Cândido Portinari	29
<i>Daniel Pacheco de Queiroz, Antonio da Luz Júnior e Arnaldo de Albuquerque Araújo</i>	
Identificação automática de cromossomos humanos	33
<i>Guilherme C. Kurtz, Tiago B. Bonini, Giovani R. Librelotto, Luis A. Perles e Michele R. Sagrillo</i>	
Visão Computacional aplicada à Localização e Mapeamento Simultâneos de Robôs Subaquáticos	37
<i>Paulo Drews Jr e Silvia Botelho</i>	

Extração Automática de Características em Imagens de Impressão Digital <i>Eduardo da Silva Ribeiro, Erikson Freitas Moraes e Gilberto Luis Valente da Costa</i>	41
Extensão do Algoritmo CCV Aplicado à Recuperação de Imagens com Base no Conteúdo Aplicada à Pesquisa Guignard <i>Rodrigo Silva Oliveira, Thatyene Louise Alves de Souza Ramos, Ana Paula Brandão Lopes, Camillo Jorge Santos Oliveira e Arnaldo de Albuquerque Araújo</i>	45
Estruturas Tritree Compressão de imagens coloridas <i>Tiago Oliveira Cunha e Vânia Cordeiro da Silva</i>	49
Redução de Ghost em imagens de MR adquiridas pelo método EPI <i>Bruno C. Marini, Marcelo V. W. Zibetti e Gabriela Castellano</i>	53
Segmentação de Dentes Utilizando a Transformada Imagem-Floresta em Sistemas Biométricos <i>Elizabeth B. Barboza, Demiam O. Fonseca e Aparecido N. Marana</i>	57
Geração da Curva de Interseção em Superfícies Trianguladas <i>André Luiz Firmino Alves e Rosana Marques da Silva</i>	61
Método Híbrido de Reconhecimento Ótico de Caracteres <i>William Ivanski, Luciano Silva e Olga R. P. Bellon</i>	65
Software de Aquisição de Dados para um Sistema de Captura de Movimentos <i>Carolina A. S. Bigonha, David L. Flam, João Victor B. Gomide e Arnaldo de A. Araújo</i>	69
SMALL GPS Sistema de Mapeamento de Linhas de Ônibus e Localização do Usuário via GPS pelo Celular <i>Carlos Henrique Macchi Gasparetto e Isabel Harb Manssour</i>	73
Autenticação Biométrica: um Sistema Híbrido Combinando Informações Globais e Locais de Impressões Digitais <i>Rubisley P. Lemes, Cristiano O. Ferreira, Olga R. P. Bellon e Luciano Silva</i>	77
Fecho Convexo e o Paradigma Rotating Calipers <i>Renato de J. Manzoni e Marco A. Piteri</i>	81
Rastreamento facial em vídeo com aplicação em monitoramento e Segurança <i>Alessandro G. F. Fior, Maurício P. Segundo, , Luciano Silva e Olga R. P. Bellon</i>	85
Simulações de Movimentos Físicos em Duas Dimensões, Utilizando Framework XNA <i>Luiz Filipe F. Carneiro e Wietske Ineke Meyering</i>	89
Uma Estrutura Hierárquica para Iluminação de Cartoons <i>Allan Carlos Rocha, Leandro Carnaúba, Renata Thomaz Linz, Adelailson Peixoto e Vinícius Mello</i>	93

Deformação de Objetos 2D Utilizando Curvas de Bézier e Mínimos Quadrados Móveis	97
<i>Douglas Cedrim e Adelailson Peixoto</i>	
Estimando Cobertura Vegetal por Análise de Imagens	101
<i>Jaime Daniel Corrêa Mendes e André Vital Saúde</i>	
Inspeção Visual de Placas de Circuito Integrado com Alta Densidade de Micro-componentes	105
<i>Felipe Gomes de Oliveira e José Luiz de Souza Pio</i>	
VHuP: a Tool to Visualize Virtual Humans	109
<i>Diogo Strube de Lima, Henry Braun e Soraia Raupp Musse</i>	
Um Aplicativo para Medição do Tamanho do Grão de Materiais Metálicos	113
<i>Francisco Roger L. Caetano, Magnaura M. dos Santos e José Pinheiro Queiroz-Neto</i>	

Prefácio

Sejam bem-vindos à sexta edição do WUW 2008. O WUW - Workshop of Undergraduate Work é um evento que visa incentivar a participação de alunos de graduação na produção e divulgação de trabalhos científicos sobre temas relacionados às áreas de Computação Gráfica e Processamento de Imagens. Em 2008, foram submetidos 34 papers, sendo 29 aceitos para publicação. Cada paper foi revisado por 2 revisores e avaliado pelos chairs do WUW.

Neste ano, a exemplo do ano anterior, os autores foram convidados a apresentarem, de maneira bastante sucinta, seus trabalhos, instigando a platéia para visitação de seus posters. Esse processo poderá resultar em discussões importantes no incentivo aos jovens pesquisadores. Desejamos a todos um excelente Workshop!

Campo Grande, outubro de 2008.

José Pinheiro Queiroz-Neto e Soraia Raupp Musse

Organização do WUW SIBGRAPI 2008

Comitê de Programa

Coordenadores do Workshop of Undergraduate Work

José Pinheiro Queiroz-Neto (Centro Fed. de Educação Tecnológica do Amazonas – CEFET/AM)
Soraia Raupp Musse (Faculdade de Informática – FACIN, PUCRS, RS)

Comitê de Programa

Adolfo Bauchspies (Universidade de Brasília – UnB)
Alessandro Bicho (Universidade Federal do Rio Grande – FURG)
Alexandre Falcao (Universidade Estadual de Campinas – Unicamp)
Alexei Machado (Pontifícia Universidade Católica de Minas Gerais – PUC-Minas)
Andre Silva (Universidade Estadual de Campinas – Unicamp)
Bruno Motta de Carvalho (Universidade Federal do Rio Grande do Norte – UFRN)
David Menotti (Universidade Federal de Ouro Preto – UFOP)
Fernando Marson (Universidade Regional Integrada – Santo Ângelo)
George Cavalcanti (Universidade Federal de Pernambuco – UFPE)
Gerson Cunha (Universidade Federal do Rio de Janeiro – UFRJ)
Guillermo Camara Chavez (Universidade Federal de Minas Gerais – UFMG)
Hélio Lopes (Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio)
Helio Pedrini (Universidade Estadual de Campinas – Unicamp)
José Pinheiro Queiroz-Neto (Centro Fed. de Educação Tecnológica do Amazonas – CEFET/AM)
Julio Jacques (Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS)
Léo Magalhães (Universidade Estadual de Campinas – Unicamp)
Luiz Marcos Gonçalves (Universidade Federal do Rio Grande do Norte – UFRN)
Luiz Gonzaga da Silveira Jr (Universidade do Vale do Rio dos Sinos – Unisinos)
Luiz Henrique de Figueiredo (Instituto de Matemática Pura e Aplicada – IMPA)
Marcelo Cohen (Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS)
Marcelo Dreux (Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio)
Marcio Netto (Escola Politécnica da Universidade de São Paulo – EPUSP)
Maria Andréia Rodrigues (Universidade de Fortaleza – UNIFOR)
Marly Costa (Universidade Federal do Amazonas - UFAM)
Neucimar Leite (Universidade Estadual de Campinas – Unicamp)
Nivando Bezerra (Universidade de Fortaleza – UNIFOR)
Olga Bellon (Universidade Federal do Paraná – UFPR)
Ricardo Torres (Universidade Estadual de Campinas – Unicamp)
Rogério Caetano (Universidade Federal do Amazonas – UFAM)
Sérgio Scheer (Universidade Federal do Paraná – UFPR)
Silvio Guimarães (Pontifícia Universidade Católica de Minas Gerais – PUC-Minas)
Soraia Musse (Faculdade de Informática – FACIN, PUCRS, RS)
Thomas Lewiner (Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio)
Valguima Odakura (Universidade Federal da Grande Dourados – UFGD)

Revisores

Adolfo Bauchspies (Universidade de Brasília – UnB)
Alessandro Bicho (Universidade Federal do Rio Grande – FURG)
Alexandre Falcao (Universidade Estadual de Campinas – Unicamp)
Alexei Machado (Pontifícia Universidade Católica de Minas Gerais – PUC-Minas)
Andre Silva (Universidade Estadual de Campinas – Unicamp)
Bruno Motta de Carvalho (Universidade Federal do Rio Grande do Norte – UFRN)
David Menotti (Universidade Federal de Ouro Preto – UFOP)
Fernando Marson (Universidade Regional Integrada – Santo Ângelo)
George Cavalcanti (Universidade Federal de Pernambuco – UFPE)
Gerson Cunha (Universidade Federal do Rio de Janeiro – UFRJ)
Guillermo Camara Chavez (Universidade Federal de Minas Gerais – UFMG)
Hélio Lopes (Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio)
Helio Pedrini (Universidade Estadual de Campinas – Unicamp)
José Pinheiro Queiroz-Neto (Centro Fed. de Educação Tecnológica do Amazonas – CEFET/AM)
Julio Jacques (Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS)
Léo Magalhães (Universidade Estadual de Campinas – Unicamp)
Luiz Marcos Gonçalves (Universidade Federal do Rio Grande do Norte – UFRN)
Luiz Gonzaga da Silveira Jr (Universidade do Vale do Rio dos Sinos – Unisinos)
Luiz Henrique de Figueiredo (Instituto de Matemática Pura e Aplicada – IMPA)
Marcelo Cohen (Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS)
Marcelo Dreux (Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio)
Marcio Netto (Escola Politécnica da Universidade de São Paulo – EPUSP)
Maria Andréia Rodrigues (Universidade de Fortaleza – UNIFOR)
Marly Costa (Universidade Federal do Amazonas - UFAM)
Neucimar Leite (Universidade Estadual de Campinas – Unicamp)
Nivando Bezerra (Universidade de Fortaleza – UNIFOR)
Olga Bellon (Universidade Federal do Paraná – UFPR)
Ricardo Torres (Universidade Estadual de Campinas – Unicamp)
Rogério Caetano (Universidade Federal do Amazonas – UFAM)
Sérgio Scheer (Universidade Federal do Paraná – UFPR)
Silvio Guimarães (Pontifícia Universidade Católica de Minas Gerais – PUC-Minas)
Soraia Musse (Faculdade de Informática – FACIN, PUCRS, RS)
Thomas Lewiner (Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio)
Valguima Odakura (Universidade Federal da Grande Dourados – UFGD)

Extração de Informações de Cor e Forma para Detecção de Placas de Trânsito em Imagens

Ricardo Cezar B. Rodrigues (bolsista CAPES), Sergio Roberto M. Pellegrino (orientador),

Carlos Henrique Q. Forster (orientador)

Instituto Tecnológico de Aeronáutica - ITA

Divisão de Computação

Praça Marechal Eduardo Gomes, 50 - São José dos Campos, Brasil

{rcezar, pell, forster}@ita.br

Resumo

O reconhecimento automático de sinais de trânsito prova-se útil na concepção de sistemas de apoio à motoristas ou até mesmo de veículos autônomos. No entanto, a capacidade da visão computacional ainda é muito baixa quando comparada ao sistema de visão humana, devido a quantidade de informação que esta pode processar em tempo real e lidar com a complexidade e variação de ambientes. Desta forma, os sistemas de visão computacional reconhecedores de objetos devem utilizar técnicas que evitem processamento desnecessário. Este trabalho propõe um modelo de detecção de regiões candidatas a placas em imagens baseado em informações de cor e forma, onde o objetivo é eliminar regiões da imagem que não têm chances de conter placas de trânsito antes da fase de reconhecimento. O trabalho também apresenta um experimento com os algoritmos supervisionados k-nn e Redes bayesianas e seus resultados preliminares que atingem até 90% de acertos.

1. Introdução

As placas de transito são projetadas para se destacar em ambientes poluídos visualmente, possuindo características como forma, cor, figuras geométricas em contraste com o ambiente em que se encontram. Em geral podem ser facilmente reconhecidas por seres humanos, pois, através da percepção visual, uma pessoa é capaz de identificar um sinal de transito pelo seu contraste, já que sua atenção será muito mais despertada quanto mais contraste existir entre os estímulos.

Trabalhar com imagens digitais de placas para um sistema automático de identificação pode não ser uma tarefa trivial, pois é necessária uma análise minuciosa sobre a va-

riação de cores, iluminação, ambiente entre outros fatores. Devido a exposição das placas a diversas condições climáticas elas sofrem degradação e apresentam variação em sua coloração e até mesmo em suas formas, ver Figura 1, ou seja, um mesmo tipo de placa pode apresentar características diferentes.



Figura 1. Imagem de placa degradada.

Este trabalho propõe um modelo para detecção de placas em imagens baseado na percepção humana. Como as placas geralmente são pintadas com cores vibrantes em contraste com a cor branca ou preta, o modelo utiliza informações de cor para segmentar possíveis regiões de interesse e em seguida extraírem características de forma dos objetos segmentados. Algoritmos simples são utilizados para a classificação de regiões como candidatas ou não a placas. A seção seguinte descreve alguns trabalhos correlatos, seguida das seções experimentos e considerações finais.

2. Trabalhos Correlatos

A detecção de placas de trânsito aparece como fase inicial em muitos trabalhos de identificação de sinais de trânsito. O objetivo desta fase é evitar o processamento desnecessário em regiões da imagem que não possuem placas ou objetos de interesse.

Muitos trabalhos recentes apresentam o uso de informações de cores para identificar regiões candidatas a placa,

[1],[4]. Em [5] se apresenta um modelo de detecção de sinais de trânsito baseado na visão humana que extrai informações de cor e atributos de forma. O modelo de cores LCH (Lightness, Chroma, Hue) foi utilizado para segmentar a reunião das imagens por cor. A extração de características dos formatos externos das placas (círculo, retângulo ou triângulo) utilizou histogramas de medidas de orientações de verticalidade e horizontalidade detectadas através de uma máscara com 49 nós sensores. O Trabalho também possui um módulo de reconhecimento e utilizou 98 sinais de trânsito britânicos com taxa de acerto de 95%.

Em [3] é apresentado um módulo para o reconhecimento de sinais de trânsito baseado na detecção de formas. No módulo, retângulos são detectados através de detecção de bordas e círculos são identificados através da transformada Hough. Embora a utilização de métodos robustos como o de Hough pode ser lenta para processar grandes imagens [6], o método foi capaz de obter taxa de acertos de até 90%.

3. Experimento

Os experimentos realizados neste trabalho consistem basicamente nas etapas de construção do *dataset* e da classificação de imagens através de algoritmos supervisionados e treinados com um conjunto de amostras.

3.1. Conjunto de treinamento (*Dataset*)

O conjunto de amostras (*dataset*) para a base de aprendizagem foi construído a partir de 97 fotos de placas de trânsito com resolução 480X360 pixels. Diversas condições de iluminação foram testadas porque as imagens foram obtidas sobre diferentes níveis de incidência da luz solar num mesmo dia.

As imagens deste trabalho possuem amostras de 12 tipos de placas, porém como já dito anteriormente este trabalho não será reconhecer cada tipo de placa e sim regiões candidatas a placas, assim cada amostra será rotulada com uma das seguintes classes: círculo, triângulo, losângio ou octógono, ver Figura 2.

3.1.1. Segmentação Para este processo o primeiro passo foi eliminar ruídos da imagem, o que foi obtido com a utilização de um filtro de suavização. Após este pré-processamento um filtro de cor foi utilizado com o intuito de identificar regiões de contraste das placas, trabalhando-se com as cores vermelha e amarela, comumente presentes nas regiões de placas nas imagens.

Neste segmento do trabalho adotou-se os componentes matiz (hue) e saturação (saturation) do modelo de cores HSB, este modelo foi utilizado devido a independência entre seus canais, diferente de outros modelos como RGB [5]. Para criar o filtro de cor foram medidos os intervalos de



Figura 2. Exemplos de placas do banco de amostras.

matiz e saturação das cores vermelho e amarelo, que aparecem nas placas em parte das amostras. Este passo foi realizado com o auxílio da ferramenta IMAGEJ¹. Os intervalos encontrados podem ser visualizados na Tabela 1

	Vermelho	Amarelo
Matiz	0.43 - 0.90	0.37 - 0.98
Saturação	0.00 - 0.05 0.93 - 1.00	0.11 - 0.15

Tabela 1. Intervalo (*threshold*) para as cores amarelo e vermelho nas camadas matiz e saturação do modelo HSB.

A Tabela 1 apresenta dois intervalos de saturação para a cor vermelha, o segundo intervalo teve que ser estabelecido para algumas imagens que foram capturadas com menor incidência de luz solar sobre as placas. Este ocorrido deve-se a simples variações de iluminação causadas por sombras de objetos ou até mesmo de nuvens que obstruíram parte dos raios solares durante alguns minutos na amostragem. Apesar da variação de luminosidade a camada matiz não foi modificada devido a sua independência.

Após obter os intervalos dos valores de matiz e saturação que representam as cores vermelha e amarela. Uma aplicação na linguagem Java foi desenvolvida utilizando o pacote IMAGEJ para obter as camadas do modelo HSB baseado no *plugin color segmentation*² e filtrar apenas as informações nos intervalos encontrados na Tabela 1.

Desta forma, novas imagens foram geradas automaticamente, já segmentadas e binarizadas. Os resultados foram

1 Software de processamento de imagens em Java e código livre, <http://rsb.info.nih.gov/ij/>

2 *Plugin Color Segmentation* para o software IMAGEJ por Maria E. Barrilla, University of Birmingham, UK

satisfatórios e eliminaram grande parte de regiões nas imagens que não tinham correlação com placas. Exemplos de imagens geradas pelo filtro podem ser visualizadas na Figura 3.



Figura 3. Exemplos de saída após a segmentação por cor.

3.1.2. Extração de Atributos de forma Esta fase do trabalho teve como objetivo utilizar as imagens já segmentadas e binarizadas para extrair características que representem as formas geométricas das regiões de contorno das placas. Antes da extração ainda existe o pré-processamento realizado pelo *plugin* do IMAGEJ *shapedescriptor* [7] para identificar as bordas dos objetos segmentados, e, através dos pontos que compõem a borda de cada objeto, extrair os atributos: área, perímetro, ângulo, fator de forma, circularidade, compacticidade, proporção (*aspect ratio*), maior e menor eixo.

3.1.3. Seleção de atributos Uma análise foi realizada a fim de reduzir a quantidade de atributos que possam comprometer a interpretação dos resultados. Para tornar a caracterização dos objetos invariante à rotação o atributo ângulo que descreve o gradiente não foi utilizado no conjunto de dados, pois as imagens de placas podem ser capturadas de diferentes ângulos ou podem estar fixadas com ângulos diferentes.

O atributo área também não foi utilizado na base de dados, pois objetos que foram segmentados e não são placas podem conter áreas semelhantes à área de placas nas imagens, além disso, se as imagens forem capturadas a uma distância padrão o atributo área torna o modelo variante à escala.

3.1.4. Análise de regressão linear múltipla Uma análise de regressão múltipla com os atributos restantes foi adotada para determinar os atributos mais e menos significantes e assim reduzir o número de atributos e diminuir o tempo de

processamento. A regressão foi realizada no WEKA³ com uma configuração para seleção de parâmetros que retornou uma equação com apenas quatro atributos: Fator de Forma 1, Compaticidade 2, Proporção 3 e o próprio menor eixo.

$$\frac{4\pi \text{area}}{\sqrt{\text{perímetro}}} \quad (1)$$

$$\sqrt{\frac{4\pi}{\text{area}}} \quad \text{maioreixo} \quad (2)$$

$$\frac{\text{maioreixo}}{\text{menoreixo}} \quad (3)$$

3.2. Avaliação dos algoritmos de classificação

Os experimentos foram conduzidos com a intenção de avaliar a eficácia e eficiência de algoritmos simples de classificação utilizando os atributos de forma extraídos das regiões segmentadas por cor para a detecção de regiões candidatas a placas de trânsito durante a fase de geração do *dataset*.

3.2.1. Configurações Foram utilizadas as implementações dos algoritmos supervisionados k-nn e redes bayesianas da versão 3.5.2 do software WEKA com suas configurações padrões, detalhes sobre os algoritmos utilizados podem ser encontrados em [2]. Para cada algoritmo foi utilizada uma validação cruzada com 10 dobras do conjunto de dados para uma estimativa mais confiável da generalização do erro. Os algoritmos tiveram como entrada 147 instâncias de 5 classes que podem ser visualizadas na Tabela 2.

	N. Instâncias
Círculo	33
Octógono	7
Losango	31
Triângulo	11
Não-placa	65
Total	147

Tabela 2. Distribuição das instâncias do dataset.

3.2.2. Experimento A validade dos resultados é avaliada através da matriz de confusão e da proporção de falsos positivos e falsos negativos. A Tabela 3 apresenta os resultados de classificação, tempo de treinamento e teste dos algoritmos considerando as 5 classes do dataset.

³ Coleção de algoritmos para problemas de mineração de dados implementados em Java e código livre, www.cs.waikato.ac.nz/ml/weka/

	T. Trein.	T. Test.	Inst. Corretas
K-nn	0.0001s	0.0017s	89.79%
R. bayesianas	0.0022s	0.0011s	85.71%

Tabela 3. Tempo de treinamento e teste e percentual de instâncias classificadas corretamente pelos classificadores.

a	b	c	d	e	← classificado como
32	0	1	0	0	a = círculo
1	2	1	0	3	b = octógono
1	1	25	3	1	c = losango
0	0	0	11	0	d = triângulo
0	1	2	0	62	e = não placa

Tabela 4. Matriz de confusão do classificador K-nn.

É possível observar nas Tabelas 4, 5 pelas matrizes de confusão que grande parte das amostras da classe octógono foram classificadas incorretamente. Isto significa que as amostras desta classe não são suficientes para discriminá-la. Já as instâncias de triângulos não apresentam classificações incorretas em ambos classificadores.

Considerando-se que em uma fase posterior serão reconhecidas as regiões detectadas como placas, pode-se então relevar as instâncias que foram confundidas com outras instâncias de uma outra classe de placa, ou seja, reduzir as classes apenas à "placa" ou "não-placa". Desta forma não haveria problema em ter, por exemplo, uma classificação incorreta de uma placa triangular como circular, já que as duas são placas e possivelmente serão identificadas na fase de reconhecimento onde se levará em consideração outras características da região candidata.

a	b	c	d	e	← classificado como
30	0	2	0	1	a = círculo
3	0	1	0	3	b = octógono
0	0	24	5	2	c = losango
0	0	0	11	0	d = triângulo
2	2	0	0	61	e = não placa

Tabela 5. Matriz de confusão do classificador Redes Bayesianas.

4. Conclusões e Trabalhos Futuros

A grande quantidade de instâncias classificadas corretamente pelo algoritmos de classificação, assim como os tempos de treinamento e teste mostram que o modelo proposto permitiu obter resultados satisfatórios. O modelo de cor HSB mostrou-se eficiente e eficaz na fase de segmentação e os atributos selecionados através de regressão mostraram-se capazes e suficientes para discriminar as formas geométricas encontradas nas bordas das placas.

Como as amostras foram capturadas em um período de algumas horas num mesmo dia, este banco de amostras não caracteriza as diversas situações climáticas em que as placas poderiam estar inseridas. A variação climática pode ser um dos grandes problemas na identificação de placas pois uma imagem de uma placa em um dia ensolarado e uma outra imagem desta mesma placa em um dia chuvoso podem apresentar características como, cor e iluminação muito diferentes.

Em trabalhos futuros, pretende-se aumentar o conjunto de amostras, principalmente nas classes com pouca representatividade. Além disso obter imagens em diferentes condições climáticas para validar o segmentação pelo modelo de cor HSB para diferentes intervalos de cores em um mesmo conjunto de amostras.

Referências

- [1] V. Andrey and K. H. Jo. Automatic detection and recognition of traffic signs using geometric structure analysis. *SICE-ICASE, 2006. International Joint Conference*, pages 1451–1456, Oct. 2006.
- [2] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, 2001.
- [3] A. H. Fabien Moutarde, Alexandre Bargeton and L. Chanusso. Modular traffic signs recognition applied to on-vehicle real-time visual detection of american and european speed limit signs. *proceedings of 14th World congress on Intelligent Transportation Systems (ITS)*, pages 9–13, october 2007.
- [4] C.-Y. Fang, S.-W. Chen, and C.-S. Fuh. Road-sign detection and tracking. *Vehicular Technology, IEEE Transactions on*, 52(5):1329–1341, Sept. 2003.
- [5] X. Gao, L. Podladchikova, D. Shaposhnikov, K. Hong, and N. Shevtsova. Recognition of traffic signs based on their colour and shape features extracted using human vision models. *17(4):675–685*, August 2006.
- [6] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez-Moreno, and F. Lopez-Ferreras. Road-sign detection and recognition based on support vector machines. *Intelligent Transportation Systems, IEEE Transactions on*, 8(2):264–278, June 2007.
- [7] J. P. O. L. I. Syverud Kristin, Chinga Gary and W. Knut. Analysis of lint particles from full-scale printing trials. *Appita Journal: Journal of the Technical Association of the Australian and New Zealand Pulp and Paper Industry*, pages 286–290, july 2007.

Equalização de Histograma para Melhoria de Contraste de Imagens Digitais Codificadas pela Técnica de Quantização Vetorial

Marcelo Portela Sousa* e Marcelo S. Alencar (Orientador)

Instituto de Estudos Avançados em Comunicações

Universidade Federal de Campina Grande

Campina Grande, PB, Brasil

{marporsou,malencar}@iecom.org.br

Carlos Antônio Vieira Vasconcelos Júnior
Laboratório de Controle
Universidade Federal da Bahia
Salvador, BA, Brasil
carlosvasconcelos@ufba.br

Waslon Terlizzi A. Lopes (Orientador)
Faculdade ÁREA1FTE
Faculdade de Ciência e Tecnologia
Salvador, BA, Brasil
waslon.lopes@area1fte.edu.br

Resumo

Este trabalho apresenta resultados da compressão de imagens por meio da utilização da técnica de quantização vetorial (QV). No entanto, como a QV é uma técnica com perdas e que introduz uma distorção na imagem original, foi implementado um algoritmo em linguagem C para equalização de histograma das imagens digitais com a finalidade de minimizar os efeitos dessa distorção. Assim sendo, o objetivo do presente trabalho é avaliar os efeitos de imagens comprimidas com o uso da técnica de quantização vetorial e como a equalização pode melhorar a aparência desta imagem. Para o projeto de quantizadores vetoriais utilizou-se o algoritmo K-Means que tem boa eficiência. Os resultados são apresentados por meio de simulações que mostram os efeitos da combinação das técnicas de quantização e equalização.

1. Introdução

Uma das técnicas mais utilizadas para compressão de imagens é a quantização vetorial (QV) [1], pois possui um bom desempenho na redução do número de bits necessários para representar imagens.

O erro inserido pela QV pode ser avaliado por medidas de distorção subjetivas ou objetivas. Apesar dos dois tipos serem amplamente utilizados, as medidas objetivas possuem a vantagem de indicar pequenas diferenças que provavelmente não seriam perceptíveis com as medidas subjetivas. Uma medida objetiva muito utilizada na avaliação de desempenho de sistemas de compressão de imagens é a relação sinal-ruído de pico (PSNR – Peak-Signal-to-Noise-Ratio) que será utilizada neste trabalho para medir o desempenho da quantização [2].

Considerando a escala de níveis de cinza e uma quantidade de 8 bits, os pixels podem assumir um valor entre zero e 255, em que zero representa a cor preta e 255 representa a cor branca. É desejável para a visualização da riqueza de detalhes da imagem, que esta não seja nem predominantemente escura, nem predominantemente clara, em que o seu histograma seja bem distribuído em toda a escala. Uma das técnicas mais utilizadas para melhoria de contraste de imagens digitais é a equalização de histograma [2].

Em [3] a técnica de equalização de histograma é utilizada sobre o dicionário de um quantizador vetorial estruturado em árvore.

No intuito de diminuir os efeitos do processo de quantização na visualização da imagem, o presente trabalho propõe um novo método utilizando a técnica de equalização de histograma para diminuir a distorção originada no processo da QV e, portanto, aumentar a qualidade das imagens digitais de saídas processadas.

* Os autores gostariam de expressar os seus agradecimentos ao CNPq e à Companhia de Eletricidade do Estado da Bahia – Coelba (P&D 0047-013/2005 “Supervisão Remota por Imagem de Subestações Automatizadas e Dessassistidas”) pelo apoio financeiro e ao Eng. José Caetano Pugliese Guimarães pelas discussões e comentários que contribuíram para a conclusão deste trabalho.

2. Quantização Vetorial

A quantização vetorial [4, 5] tem sido utilizada para processamento digital de sinais em diversos sistemas de codificação, permitindo a obtenção de elevadas taxas de compressão [6]. A QV é definida como sendo o mapeamento Q de um vetor de entrada x pertencente ao espaço Euclidiano K -dimensional, \mathbb{R}^K , em um vetor pertencente a um subconjunto finito \mathcal{W} de \mathbb{R}^K , ou seja

$$Q : \mathbb{R}^K \rightarrow \mathcal{W}. \quad (1)$$

O conjunto $\mathcal{W} = \{w_i; i = 1, 2, \dots, N\}$ é chamado de dicionário e consiste no conjunto de vetores chamados de vetores-código; K é a dimensão do quantizador vetorial e N é o tamanho do dicionário, isto é, o número de vetores-código. O mapeamento Q introduz um particionamento de \mathbb{R}^K em N células (denominadas regiões de Voronoi) $\{S_i; i = 1, 2, \dots, N\}$, [7] tais que

$$\bigcup_{i=1}^N S_i = \mathbb{R}^K \text{ e } S_i \cap S_j = \emptyset \text{ para } i \neq j, \quad (2)$$

em que cada célula S_i é definida por

$$S_i = \{x : Q(x) = w_i\}, \quad (3)$$

sobre o mapeamento dos vetores de entrada no i -ésimo vetor de reprodução [8]. Mais precisamente,

$$S_i = \{x : d(x, w_i) < d(x, w_j), \forall j \neq i\}, \quad (4)$$

onde $d(\cdot, \cdot)$ denota uma medida de distorção.

A seleção ótima dos vetores minimiza o valor do erro de reconstrução de imagens de acordo com o critério de distorção esperado [8]. Em um processo de codificação de um sinal da fonte, geralmente utiliza-se uma seqüência de vetores, usando um dicionário. O processo de codificação é realizado substituindo cada vetor de entrada com o índice ao vetor do dicionário que o representa melhor. Para reconstruir o sinal, cada índice codificado é substituído com o vetor apropriado do dicionário. Como cada vetor no sinal reconstruído não é uma réplica exata do vetor original de entrada, a quantização vetorial pode introduzir algum erro (ou distorção) no sinal [6, 9]. A taxa de codificação, que fornece o número de *bits* por componente do vetor é dada por

$$R = \frac{1}{k} \log_2 N. \quad (5)$$

Em se tratando de codificação de imagens, R é a quantidade de *bits* por *pixel*.

Neste trabalho foram implementados algoritmos em linguagem C para a codificação dos índices dos vetores de entrada. Para o projeto de dicionários foi utilizado o algoritmo *K-means* [10, 11].

3. Equalização de Histograma

Os níveis de cinza em uma imagem podem ser vistos como quantidades aleatórias no intervalo $[0, 1]$. Considerando r como a variável aleatória contínua associada à ocorrência desses níveis, então $p_r(r)$ é a função densidade de probabilidade de r [12].

A equalização de histograma para melhoria de contraste é baseada na uniformização da função densidade de probabilidade dos níveis de cinza da imagem, representados pela variável r . A função transformadora, $s = T(r)$, que possibilita essa modificação corresponde à função cumulativa de probabilidade (FCP) de r , em que as seguintes condições devem ser satisfeitas:

- (a) $T(r)$ é uma função injetora e monotonicamente crescente no intervalo $0 \leq r \leq 1$ e
- (b) $0 \leq T(r) \leq 1$ para $0 \leq r \leq 1$.

Em se tratando de processamento digital de imagens, os conceitos apresentados devem ser formulados no domínio discreto. Desta forma, valores de cinza com níveis discretos estão associados com as probabilidades [1]:

$$p_r(r_k) = \frac{n_k}{n}, \quad 0 \leq r_k \leq 1 \text{ e } k = 0, 1, \dots, L-1 \quad (6)$$

em que L é o número de níveis de cinza, $p_r(r_k)$ é a probabilidade do k -ésimo nível de cinza, n_k é o número de vezes que esse nível de cinza aparece na imagem e n é o número total de *pixels* da imagem. O gráfico de $p_r(r_k)$ em função de r_k é chamado de *histograma* e a técnica usada para obter um histograma uniforme é conhecida como *equalização* ou *linearização* de histograma [2].

A forma discreta da função de transformação é dada pela relação

$$\begin{aligned} s_k &= T(r_k) = \sum_{j=0}^k \frac{n_j}{n} \\ &= \sum_{j=0}^k p_r(r_j) \quad 0 \leq r_k \leq 1 \text{ e } k = 0, 1, \dots, L-1. \end{aligned} \quad (7)$$

A transformação inversa é denotada por [1]

$$r_k = T^{-1}(s_k) \quad 0 \leq s_k \leq 1, \quad (8)$$

em que $T(r_k)$ e $T^{-1}(s_k)$ satisfazem as Condições (a) e (b) apresentadas anteriormente. A função de transformação $T(r_k)$ pode ser calculada diretamente da imagem usando a Equação 7.

4. Resultados

Nesta seção são apresentados os resultados obtidos de simulações referentes a quantização e equalização de histograma da imagem de teste Lena. A imagem foi codificada

no formato PPM com 256×256 pixels e cada componente de cor foi codificada a 8 bits. As três componentes de cor (RGB), no formato PPM, podem ser facilmente acessadas devido ao fato de serem apresentadas de forma alternada no arquivo ASCII. A decomposição das componentes RGB e a equalização de histograma foi implementada em linguagem C, em que a equalização foi aplicada em cada componente separadamente. As Figuras 1(b) e 2 mostram os resultados da equalização da imagem Lena e das suas componentes RGB, respectivamente.

A QV foi considerada com a dimensão do quantizador $K = 16$ (blocos 4×4 pixels) e o tamanho do dicionário $N = 512$ vetores-código. A taxa de codificação correspondente, portanto, foi de $R = 0,5625$ bpp. O algoritmo utilizado para o projeto dos dicionários foi o *K-Means*, usando seis imagens de treino com 256×256 pixels. Os algoritmos foram executados até que a modificação na distorção introduzida ao se representar os vetores de treino pelo dicionário fosse inferior a 0,1% ($\epsilon = 0,001$). A Figura 3(b) mostra os resultados da quantização da imagem Lena. Para avaliação dessa imagem processada utilizou-se a PSNR. O cálculo da PSNR para a imagem Lena reconstruída após a QV resultou em um valor de 41,94 dB.

Após a aplicação das técnicas de QV e equalização separadamente, as duas técnicas foram combinadas. Inicialmente, a equalização foi aplicada na imagem Lena original e em seguida a imagem processada foi comprimida utilizando a técnica de QV. Na Figura 4 é possível observar os efeitos desta aplicação, em que foi obtido um valor de $PSNR = 28,50$. Comparando com o resultado obtido anteriormente percebe-se que a equalização diminuiu a PSNR da imagem reconstruída. Esse fato ocorre devido a equalização de histograma inserir uma distorção na imagem. Analisando subjetivamente, percebe-se que a equalização provoca uma melhoria no contraste da imagem. Para uma avaliação objetiva da equalização é recomendado utilizar o desvio padrão global ou em janelas na imagem.

A Figura 5 mostra os efeitos da aplicação da equalização à imagem quantizada. A equalização melhora o contraste da imagem, porém quando comparada com a equalização da imagem antes da aplicação da QV, apresenta uma maior distorção. Este resultado é esperado pois a imagem, equalizada e quantizada, é submetida a dois processamentos consecutivos o que provoca um aumento na distorção da imagem.

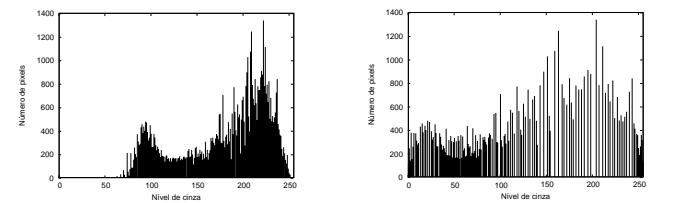
5. Conclusão

Este artigo apresentou duas técnicas para processamento digital de imagens: A QV utilizada para compressão e a equalização de histograma utilizada para melhoria de contraste. As imagens de saída processadas após a QV apresentaram níveis de distorção aceitáveis, o que implica em

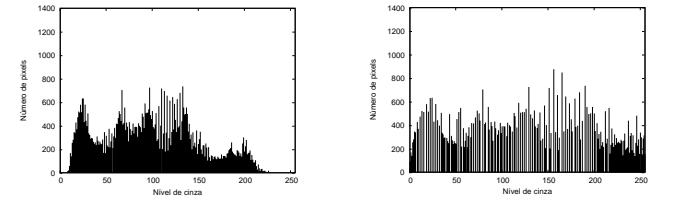


(a) Imagem Original. (b) Imagem Equalizada.

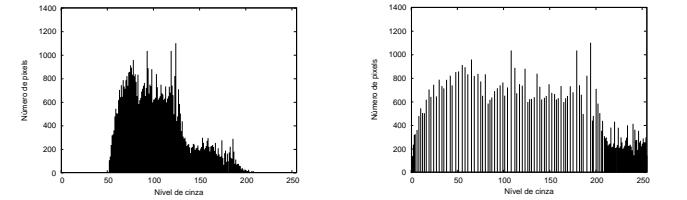
Figura 1. Equalização da Imagem Lena.



(a) Componente R: histograma original.
(b) Componente R: histograma equalizado.



(c) Componente G: histograma original.
(d) Componente G: histograma equalizado.



(e) Componente B: histograma original.
(f) Componente B: histograma equalizado.

Figura 2. Equalização da Imagem Lena.



(a) Imagem Original.

(b) Imagem Quantizada.

Figura 3. Quantização da Imagem Lena.



(a) Imagem Original.

(b) Imagem Equalizada e Quantizada.

Figura 4. Equalização e Quantização da imagem Lena.



(a) Imagem Original.

(b) Imagem Quantizada e Equalizada.

Figura 5. Quantização e Equalização da imagem Lena.

uma boa taxa de compressão. Para a equalização de histograma, as imagens processadas tiveram sua aparência melhorada, de modo que é possível observar que a equalização torna os histogramas bem distribuídos na escala de cinza.

Os resultados das simulações mostram que a aplicação da técnica de equalização de histograma diminui a PSNR quando aplicada em conjunto com a QV, entretanto, o efeito da melhoria do contraste das imagens continua sendo observado. Contudo, a aplicação da equalização antes da QV apresentou uma menor distorção. Como trabalhos futuros, os autores pretendem investigar e elaborar um método ainda mais eficiente para compressão de imagens, com a combinação do método proposto e outros métodos de quantização vetorial.

Referências

- [1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, New York, 1 edition, 1992.
- [2] W. T. A. Lopes, *Diversidade em Modulação Aplicada à Transmissão de Imagens em Canais com Desvanecimento*, Tese de Doutorado, Universidade Federal de Campina Grande, Campina Grande, PB, Brasil, Junho 2003.
- [3] P. C. Cosman, R. M. Gray, and E. A. Riskin, “Combining vector quantization and histogram equalization,” in *Proceeding of the Data Compression Conference (DCC'91)*, April 1991, pp. 113–118.
- [4] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, MA, 1992.
- [5] R. M. Gray, “Vector quantization,” *IEEE ASSP Magazine*, vol. 1, no. 2, pp. 4–29, April 1984.
- [6] F. J. A. Oliveira, J. P. F. C. Freire, A. L. O. Cavalcanti Jr., E. N. Arcos Neto, P. C. M. L. Albuquerque, W. T. A. Lopes, and F. Madeiro, “Aceleração do algoritmo k -means,” in *Anais do XV Congresso Brasileiro de Automática (CBA'2004)*, Gramado, RS, Brasil, Setembro 2004.
- [7] A. Gersho, “On the structure of vector quantizers,” *IEEE Transactions on Information Theory*, vol. 28, pp. 157–166, March 1982.
- [8] J. Makhoul, S. Roucos, and H. Gish, “Vector quantization in speech coding,” *Proceedings of the IEEE*, vol. 73, pp. 1551–1558, November 1985.
- [9] M. A. L. Perna, “Compactação de imagens digitais,” in *Anais do XVII Congresso Brasileiro de Cartografia (XVII SBC)*, Salvador, BA, 1995.
- [10] Y. Linde, A. Buzo, and R. M. Gray, “An algorithm for vector quantizer design,” *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, January 1980.
- [11] Yih. C. Lin and S. C. Tai, “A fast Linde-Buzo-Gray algorithm in image vector quantization,” *IEEE Transactions on Circuits and Systems*, vol. 45, pp. 432–435, March 1998.
- [12] S. Carrato, “Image vector quantization using ordered codebooks: Properties and applications,” *Signal Processing*, vol. 40, pp. 87–103, 1994.

Um Novo Algoritmo para Detecção de Contorno de Melanomas em Imagens de Pele

da Silva, M.G.N.M.
UFPE, Eng. Biomédica
(graduação)
gomes.mariu@gmail.com

Candeias, A.L.B.
UFPE - DECART,
Dept. de Eng. Cartográfica
analucia@ufpe.br

Resumo

O melanoma é uma mancha que geralmente apresenta-se com coloração preta, castanha, vermelha, azul ou próximo da cor da pele. Pode surgir da pele sã ou de lesões pigmentadas pré-existentes. As principais diferenças entre o melanoma benigno e o maligno para um diagnóstico visual, não invasivo, são: a forma, a borda a cor e o diâmetro. Este trabalho propõe um novo algoritmo que detecta o contorno de melanomas, possibilitando assim uma melhor distinção entre os tipos benignos (borda regular) e malignos (borda irregular). Para imagens com interferência de cabelos foi criado um algoritmo simples e de fácil implementação para a retirada desses cabelos. Para auxiliar nesse diagnóstico foi criada uma ferramenta contendo esses dois algoritmos, incluindo também outras formas de visualização das imagens.

1. Introdução

O melanoma cutâneo é um tipo de câncer que tem origem nos melanócitos (células produtoras de melanina, substância que determina a cor da pele) e predominância em adultos brancos. Embora represente um percentual pequeno dos tipos de câncer de pele, o melanoma é o mais grave devido à sua alta possibilidade de metástase [3] e [4].

Quando o melanoma deixa de ser plano, formando lesão elevada na pele, é sinal de que também está progredindo em profundidade. A profundidade e a espessura são os parâmetros que definem a gravidade da lesão.

Na área da dermatologia existe uma demanda crescente no uso de técnicas de processamento digital de imagens de pele contendo lesões a serem diagnosticadas. O primeiro passo para um diagnóstico é a detecção da área lesionada. Um dos obstáculos a esse processamento é que, normalmente, as imagens de

pele possuem fios de cabelos que atrapalham a diferenciação entre segmentos de pele saudáveis e lesionados.

Existem melanomas benignos e malignos vide Figura 1.

Na Tabela 1 há uma classificação do tipo de melanoma segundo suas características.

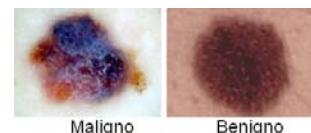


Figura 1. Imagens de melanomas benigno e maligno.

Tabela 1. Classificação de melanomas segundo suas características.

	Benigno	Maligno
Forma	Simétrica	Assimétrica
Borda	Regular	Irregular
Cor	Uniforme	Variada
Diâmetro	< 6mm	> 6mm

A ferramenta aqui proposta foi concebida com o intuito de auxiliar o diagnóstico de melanomas, baseado em imagens de pele.

Neste trabalho foi implementado um algoritmo de segmentação para se obter o contorno dos melanomas, podendo assim diferenciar melhor os melanomas benignos (borda regular) dos malignos (borda irregular). Para as imagens com muitos cabelos escuros na frente do melanoma foi implementado um algoritmo com a tentativa de retirar os cabelos para que se conseguisse fazer o contorno e saber a área do melanoma. Esse processamento é feito para evitar que se tenha que depilar o local lesionado para melhor visualização do melanoma, pois o local já está comprometido e não seria interessante irritá-lo ainda mais, o que causaria um desconforto e ainda traria incômodos estéticos. O uso de um algoritmo para realizar essa tarefa é indiscutivelmente mais prático e

tem a vantagem de realizar cálculos que podem ajudar um algoritmo de diagnóstico automático das lesões na pele.

Para isso foram feitos testes com 37 imagens sem e com pouca interferência de cabelos e 3 imagens com muita interferência de cabelos.

Na seção 2 é apresentada a metodologia, enquanto na seção 3 os resultados do novo algoritmo, na seção 4 é feita uma comparação com algoritmos já existentes na literatura e finalmente na seção 5 algumas conclusões.

2. Metodologia

Nesta seção serão descritos os passos realizados pelo algoritmo de contorno aqui proposto. A teoria de limiarização, filtragem e segmentação foram extraídas a partir de [1] e [2].

O primeiro passo é transformar a imagem original (RGB) em sua versão em tons de cinza (vide Figuras 2a e 2b) para sua posterior segmentação.

A segmentação da área do melanoma é feita a partir da imagem em tons de cinza, definindo-se limiares em seu histograma para a binarização da imagem.

Antes da limiarização, a imagem é suavizada utilizando-se um filtro de mediana 8x8 (vide Figura 2c).

Seu uso é essencial para o sucesso da segmentação. Os fatores que justificam tal filtragem são:

- As cores que aparecem no melanoma, em geral, não são uniformes;
- A possibilidade da interferência de pêlos (raros).

A opção pela filtragem da mediana se dá pelo fato desta não alterar tanto a forma do melanoma quanto um filtro de média móvel.

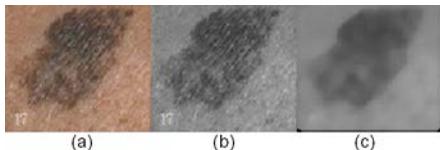


Figura 2. Imagens (a) original, (b) em tons de cinza e (c) resultado da filtragem da mediana 8x8.

Agora, analisa-se o histograma da imagem resultante (vide Figura 3), na busca de um limiar para sua binarização.

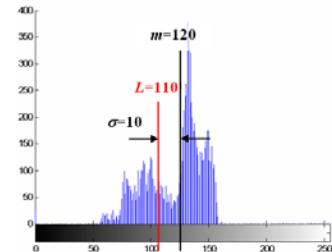


Figura 3. Histograma da imagem da Figura 2c.

Em imagens de melanomas, a maior parte da imagem é ocupada por pontos pertencentes à pele sadia. Esta por sua vez, tem um aspecto uniforme. Com base nestas informações é proposto o seguinte limiar:

$$T = m - \sigma$$

onde m é a média e σ é o desvio padrão dos tons de cinza existentes na imagem.

Determinado o limiar, toma-se a seguinte decisão: os pontos que têm tons de cinza menores que T , serão ditos pertencer à área do melanoma, os demais serão classificados como pontos da pele sadia. No histograma apresentado na Figura 3 estão destacados a média, o desvio padrão e o limiar correspondente à imagem da Figura 2c. Na Figura 4a é mostrada a imagem binarizada utilizando-se o limiar encontrado. Os pontos pretos destacam a área de pele sadia, já os brancos, o melanoma.

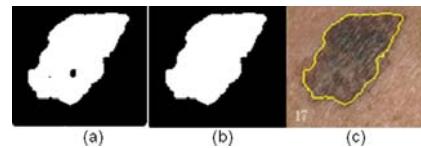


Figura 4. Imagens (a) binarizada, (b) com apenas o maior objeto e (c) contorno do objeto.

É possível observar na imagem da Figura 2c, que após a filtragem da mediana, ocorreu o aparecimento de uma borda escura. Este fato faz com que apareça, na imagem binarizada (Figura 4a), uma borda branca. Os pontos dessa borda, certamente não pertencem à área do melanoma, dessa forma, esta passa a ser preta. Outro problema que pode surgir é a aparição de mais objetos, além da área do melanoma e da borda. Isso pode ocorrer devido à existência de pêlos remanescentes da filtragem da mediana, por exemplo. Este último problema é resolvido considerando-se apenas o objeto de maior área (o melanoma), os demais, são setados para preto.

O interesse é encontrar o contorno da área do melanoma, assim as partes pretas internas ao objeto, agora único, são desprezadas, ou seja, setadas para branco (vide Figura 4b). Por fim, baseado na imagem da Figura 4c, pode-se determinar o contorno da área do melanoma na imagem original (vide Figura 4c).

A solução apresentada aqui assumiu que o melanoma é mais escuro que a pele. Entretanto, há casos em que a pele é mais escura.

Analizando-se a Figura 5a, é possível constatar que para melanomas mais claros que a pele, o histograma da imagem em tons de cinza apresenta-se invertido em relação à situação oposta (vide Figura 3).

Através de uma análise simples, é possível determinar se a pele é mais clara que o melanoma e vice-versa. Para isso calcula-se a média e a moda do histograma da imagem em tons de cinza. Se a primeira for maior que a segunda, então pode-se dizer que o melanoma é mais claro que a pele, caso contrário a pele é mais clara. Quando o primeiro caso ocorrer, deve-se, tomar o negativo da imagem em tons de cinza (vide Figura 5b), antes de aplicar o algoritmo proposto.

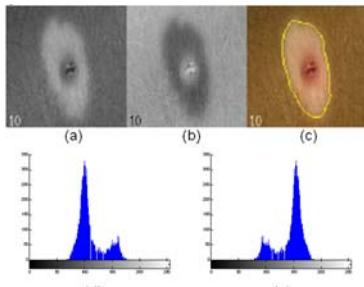


Figura 5. (a) Versão em tons de cinza da imagem original, (b) imagem (a) com as cores invertidas, (c) imagem original com o contorno detectado, (d) e (e) histogramas das imagens (a) e (b), respectivamente.

Para as imagens com muita interferência de cabelos foi implementado um algoritmo para retirá-los. A imagem original (vide Figura 6a) foi separada nas bandas R, G e B, e cada banda foi filtrada com um filtro de mediana 40×1 formando imagens que vão servir para substituir os pêlos posteriormente. A imagem original foi transformada em sua versão em tons de cinza (vide Figura 6b) e depois foi binarizada através da limiarização, com o intuito de deixar como objeto só os cabelos, como pode-se observar na Figura 6c. Em seguida os cabelos foram substituídos por outros níveis de cinza, referentes as bandas filtradas (vide Figura 6d), em seguida a imagem foi filtrada com um filtro de mediana 8×8 (vide Figura 6e).

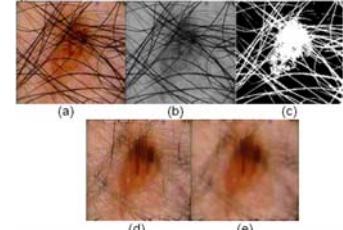


Figura 6. Imagens (a) original, (b) em tons de cinza, (c) binarizada, (d) após substituição dos cabelos e (e) filtrada.

Foi implementada uma ferramenta executável no MATLAB® para fazer processamento de imagens de melanoma, na qual estão presentes os novos algoritmos aqui apresentados e mais algumas formas de visualização da imagem.

3. Resultados

Os resultados são apresentados nas Figuras 7 e 8, onde mostram, respectivamente, resultados do novo algoritmo de contorno aplicado a diversos tipos de melanomas e resultados com imagens com interferência de cabelos.



Figura 7. Resultados do novo algoritmo de contorno.

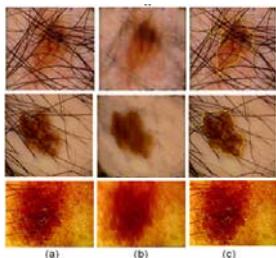


Figura 8. Imagens com interferência de cabelos, (a) imagem original, (b) resultado do algoritmo de retirada de cabelos, (c) contorno dos melanomas.

4. Comparação de algoritmos

Uma solução para o problema da interferência dos cabelos foi proposta por [7] que desenvolveu um software, chamado DullRazor, para remover fios de cabelos de imagens de pele usando técnicas de segmentação de imagens. Esse algoritmo retira melhor os cabelos como mostra a Figura 9, mas a parte de contorno do melanoma não é tão satisfatório quanto o apresentado no presente trabalho, como mostra a Figura 10.

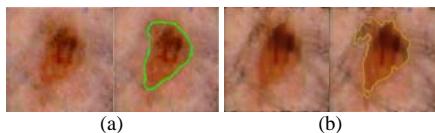


Figura 9. Algoritmo de retirada dos cabelos e algoritmo de contorno, (a) Algoritmo de DullRazor, (b) Novo Algoritmo.

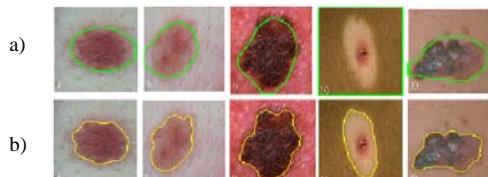


Figura 10. Algoritmos de contorno, (a) Algoritmo de DullRazor, (b) Novo Algoritmo.

Existe uma implementação do algoritmo DullRazor no MATLAB®, desenvolvido por Alexandre Zaghetto e Pedro de Azevedo Berger [6], que remove bem os cabelos, porém é insatisfatório o algoritmo de contorno, por não ser algo contínuo, como mostra a Figura 11.



Figura 11. Resultado do Algoritmo DullRazor

5. Conclusão

Este trabalho apresentou uma nova abordagem para detecção de contorno de melanomas em imagens de pele utilizando a plataforma MATLAB® para auxiliar o diagnóstico de melanomas, baseado em imagens de pele.

Foi implementado um algoritmo de segmentação para se obter o contorno dos melanomas, para diferenciação entre melanomas benignos e malignos.

Um problema enfrentado foi a retirada de cabelos por filtros para se conseguir fazer o contorno e saber a área do melanoma. Com isto evita-se de depilar o local lesionado para melhor visualização do melanoma.

Visto o que existe na literatura [5] e [7] sobre o assunto conclui-se que o resultado apresentado pode auxiliar bastante na definição do contorno destes melanomas.

Referências

- [1] Gonzalez, R.C. and Woods, R.E. *Digital Image Processing*, 2nd. Edition, 2000, Prentice Hall, Upper Saddle River, NJ.
- [2] Gonzalez, R.C., Woods, R.E. and Eddins, S. L. *Digital Image Processing using Matlab*, 2004, Prentice Hall, Upper Saddle River, NJ.
- [3] Gallagher, R.; Spinelli, J. and Lee, T. *Tanning Beds, Sunlamps, and Risk of Cutaneous Malignant Melanoma*, *Cancer Epidemiology Biomarkers & Prevention* Vol. 14, 562-566, March 2005.
- [4] Melanoma – Skin cancer reviewed – URL:<http://melanoma.blogspot.com/> Acessado em: 01/07/2008.
- [5] Nunes, E. O. and Conci, A. *Segmentação por Textura e Localização do Contorno de Regiões em Imagens Multibandas*. Artigo publicado no *IEEE Latin America Transactions*, Vol.5, NO.3, em junho de 2007.
- [6] Implementação do Algoritmo DullRazor no MATLAB – URL:www.ene.unb.br/~juliana/cursos/pimagens/projetos/aluinos/pedro_alexandre/index.html . Acessado em: 01/07/2008.
- [7] Lee, T; Ng, V.; Gallagher, R.; Coldman A. and McLean D.. Dullrazor®: A software approach to hair removal from images. *Computers in Biology and Medicine*, Volume 27, Issue 6, November 1997, Pg. 533-543.

Método de Classificação de Pontos em Esqueletos de Resolução Aumentada

Diógenes Corrêa Canoza*, André Vital Saúde (orientador)
Universidade Federal de Lavras
DCC-UFLA, CP 3037, CEP 37200-000, Lavras-MG, Brasil
dcanoza@comp.ufla.br, saude@dcc.ufla.br

Resumo

O esqueleto é uma representação simplificada de um objeto digital 3D, mas que mantém certas características do objeto original. Há propostas para a identificação dos pontos do esqueleto 3D como pontos de borda, superfície, curva ou junção. Tal identificação foi utilizada na literatura para filtrar o esqueleto 3D, facilitando a análise de forma do objeto. Recentemente foram introduzidos esqueletos em resolução aumentada, onde a classificação dos pontos de resolução normal não pode mais ser aplicada. Esta é uma das razões para que esses esqueletos até então não tenham sido utilizados em nenhuma aplicação prática. Neste trabalho é proposto um método para que a mesma classificação utilizada anteriormente em resolução normal seja estendida para a resolução aumentada. Assim, torna-se possível reproduzir os métodos de análise de forma presentes na literatura usando esqueletos de resolução aumentada.

1. Introdução

O esqueleto é uma representação simplificada de um objeto digital 3D, mas que mantém certas características do objeto original. Malandain et al. [4] propõem uma identificação dos pontos do esqueleto 3D como pontos de borda, superfície, curva ou junção. Tal identificação permite que o esqueleto 3D possa ser filtrado para facilitar a análise de forma do objeto, assim como proposto por [2].

Recentemente foram introduzidos por [3] os esqueletos em resolução aumentada. Esses esqueletos até então não foram utilizados em nenhuma aplicação prática. Com o intuito de reproduzir a descrição de [2] nos esqueletos em resolução aumentada, antes é necessário que se defina as classificações dos pontos para essa resolução. Neste trabalho é proposto um método para que a mesma classificação

descrita por [4] em resolução normal, seja estendida para a resolução aumentada.

É demonstrada a classificação dos pontos e após isto a seleção dos pontos de superfície, a partir de um esqueleto de uma imagem do hipocampo humano em resolução aumentada, mesma imagem foi utilizada por [2] em resolução normal, e os resultados da seleção dos pontos de superfície foram, visualmente, muito semelhantes aos encontrados por [2].

2. Complexos abstratos e esqueletos em resolução aumentada

O esqueleto em resolução aumentada obtido em [3] está definido no contexto dos complexos abstratos, e é baseado em um framework de esqueletização proposto por [1]. Os complexos abstratos são conjuntos totalmente ordenados que apresentam boas propriedades para a implementação de filtros de processamento de imagens que precisem preservar a topologia do objeto. O esqueleto em resolução aumentada que foi estudado está representado como um complexo, e por isso foi necessário estudar características de forma neste contexto.

E quando uma imagem é convertida em resolução aumentada cada voxel corresponde a 27 voxels totalmente ordenados, agora chamados de faces. Observe a *Figura 1*, esta é a representação de um voxel convertido para a resolução aumentada. Se houvessem dois voxels um ao lado do outro, por exemplo, eles compartilhariam algumas faces. A ordem estabelecida, baseada na *Figura 1*, da seguinte maneira:

- Uma face de ordem 3, a 3-face, está representada pelo cubo central.
- As faces de ordem 2, a 2-face, estão representadas pelos paralelepípedos que são as faces do cubo.
- As faces de ordem 1, a 1-face, estão representadas pelos paralelepípedos que são as arestas do cubo.
- As faces de ordem 0, a 0-face, estão representadas pelas esferas que são os vértices do cubo.

* Agradecimento a FAPEMIG pelo patrocínio do projeto.

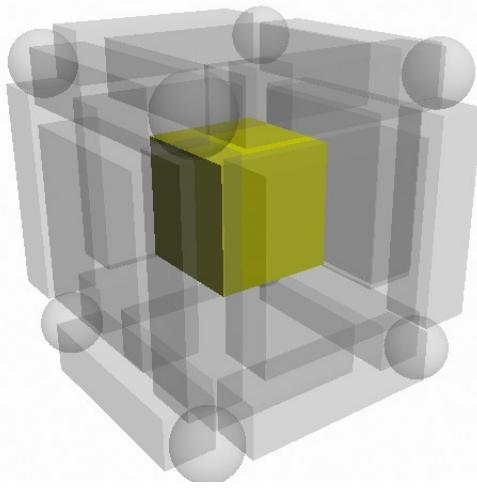


Figura 1. Representação de um 3-face com todos os 2-faces, 1-faces e 0-faces contidos por ele.

Assim sendo cada face contém todas as outras faces com ordem menor e são contidas por todas as de ordem maior, desde que estejam na vizinhança da face observada, conceito que será tratado mais a frente.

O conceito de **fechamento** diz que para uma determinada face f , devem existir todas as faces que são inclusas por f . Finalmente complexo abstrato é o conjunto de todas as faces, tal que todas estejam fechadas.

2.1. Métodos de identificação de pontos em resolução normal

Os pontos de uma superfície mediana podem ser identificados como pontos de borda, junção, superfície e curva, usando a classificação de [4]. Tal classificação utiliza-se de conceitos de conectividade 26 ou 26-vizinhos que são os 26 voxels mais próximos. A *Tabela 1* demonstra como é feita a classificação de acordo com a quantidade de componentes conexos, grupo de voxels conexos, tanto do objeto quanto do seu complementar ou fundo.

2.2. Conceitos aplicados

Para entender a identificação dos pontos em resolução aumentada é necessário entender alguns conceitos. Considerando um conjunto finito das faces X , a face $f \in X$ e se não houver nenhum $g \in X$ tal que f esteja incluso estritamente em g . Neste caso f é chamado uma **face principal**, ou um **facet**, [3].

Dizemos que f é uma **face essencial** se f for precisamente a interseção de todas as faces principais de X que

Ponto isolado	$B = 1$	$O = 0$
Ponto interior	$B = 0$	$O = 1$
Ponto de Borda	$B = 1$	$O = 1$
Ponto de Curva	$B = 1$	$O = 1$
Ponto de Junção de curva	$B = 1$	$O > 1$
Pontos de Superfície	$B = 2$	$O = 1$
Ponto de Junção Curva-Superfície	$B > 2$	$O \geq 2$
Ponto de Junção de Superfícies	$B > 2$	$O = 1$
Ponto de Junção Curva-Superfície	$B > 2$	$O \geq 2$

Tabela 1. Tabela de classificação de pontos.

B: número de componentes conexos do complementar (Background). **O:** número de componentes conexos do objeto (Object).

contêm f . Denotamos por $\text{Ess}(X)$ o conjunto composto de todas as faces essenciais de X . As faces essenciais têm contexto sobre topologia, que não é tratada nesse trabalho, mas foi observado que sua utilização é vital para os resultados.

2.3. Utilizando vizinhança nos complexos abstratos

Como visto, há uma relação de ordem entre as faces nos complexos abstratos e a partir desta ordem existe uma conectividade própria para cada face, que são todas as faces que contém ou são contidas pela face observada. Então para que se possa utilizar o conceito de vizinhança nos complexos abstratos antes se faz necessário observar qual a ordem da face. Assim podemos enumerar as vizinhanças de cada ordem : uma 3-face reconhece 26 vizinhos pois contém todas as faces de ordem menor a sua volta, assim como a 0-face que é contida por todas as outras faces então enxerga todas as faces a sua volta. A 2-face reconhece como vizinho as duas 3-face que a contém, as quatro 1-faces e quatro 0-faces que são contidas por ela, então possui uma vizinhança 10. E a 1-face reconhece como vizinho as duas 0-faces contidas por ela, as quatro 2-faces e quatro 3-face que a contém, então possui uma vizinhança 10. Uma última consideração é que nenhuma face é vizinha de outra face de mesma ordem.

3. Identificação de pontos em resolução aumentada

Primeiramente recebemos de entrada uma imagem nos complexos abstratos no caso a *Figura 2*. Depois para que se possa realizar a identificação dos pontos em resolução aumentada é necessária a remoção das faces não essenciais, obtendo a *Figura 3*. Isso se faz através da marcação de todas as faces que são essenciais, já definidas anteriormente,

depois se percorre a imagem desconsiderando as faces não marcadas. Em seguida percorre-se a imagem selecionando cada face do objeto e avaliando qual sua ordem, para que se possa identificar a vizinhança a ser observada. A partir das faces dos objetos existentes na vizinhança da face selecionada é possível enumerar a quantidade de componentes conexos do objeto. Essa enumeração deve ser feita considerando a face avaliada como não sendo parte dos componentes conexos. Para identificação também é necessário encontrar os componentes conexos do fundo. Para isso basta aplicar a mesma técnica considerando o complementar do objeto, observando a partir da mesma face. E mais uma vez deve-se desconsiderar a face avaliada.

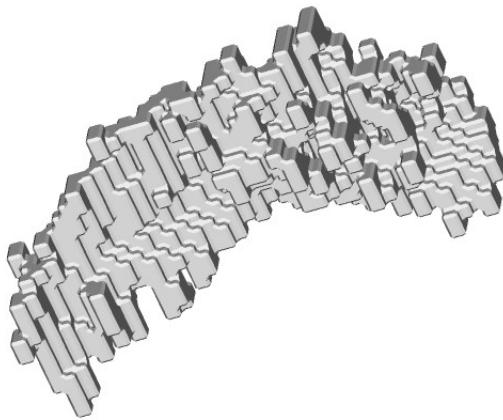


Figura 2. Esqueleto em resolução aumentada do hipocampo humano.

Uma vez encontrados os componentes conexos do fundo e do objeto pode-se usar a *Tabela 1* para classificar a face. E assim deve ser feito para cada uma das faces da imagem. Esta técnica é uma adaptação da descrita por [4].

Com todas as faces classificadas é possível separar todas as faces de superfície como demonstrado na *Figura 4*.

Os passos para essa classificação e obtenção dos pontos de superfície e junção de superfície são descritos no algoritmo abaixo. No algoritmo, a função *RemEss()* recebe uma imagem e a retorna com todas as faces que não são essenciais removidas, a função *ClassificaPontos()* recebe uma imagem e a retorna com todos os pontos classificados através do método descrito anteriormente e a função *PontosSuperficie()* recebe uma imagem já com os pontos classificados e retorna o conjunto dos pontos de superfície e junção de su-

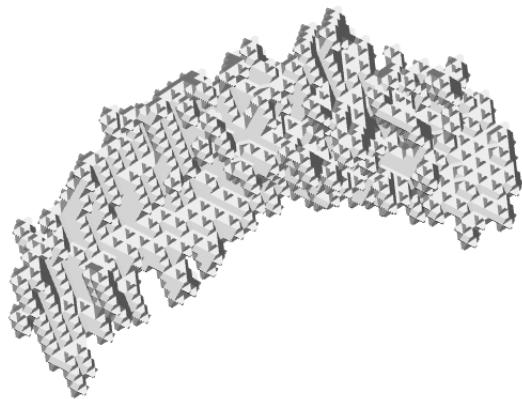


Figura 3. Faces essenciais da Figura 2.

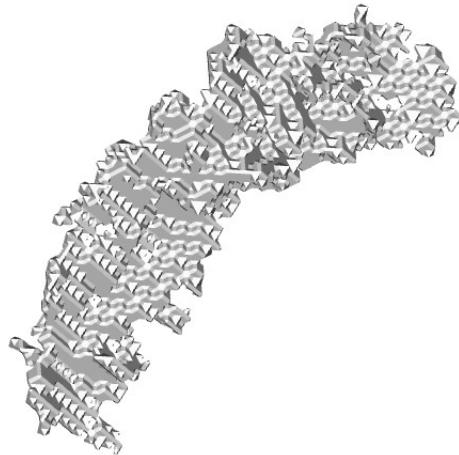


Figura 4. Faces classificadas como sendo de Superfície ou junção de superfície a partir da Figura 2.

perfície do objeto.

4. Resultados

Como resultado conseguiu-se extrair todos os pontos de superfície e junção de superfície em um esqueleto Euclidiano em resolução aumentada através da criação das regras para classificação dos pontos em resolução aumentada e de um algoritmo que a implementa.

5. Conclusões

Concluiu-se que a classificação dos pontos em resolução aumentada e com a possibilidade de se extrair as faces

input : Uma Imagem Im descrita no contexto dos complexos abstratos.
output : Conjunto S com todos os pontos de superfície e junção de superfície de Im .
1 $S \leftarrow \text{RemEss}(Im);$
2 $S \leftarrow \text{ClassificaPontos}(S);$
3 $S \leftarrow \text{PontosSuperficie}(S);$

Algorithm 1: Algoritmo de seleção de superfície

de superfícies da imagem é um passo importante. Pois a imagem analisada é uma imagem de um esqueleto do hipocampo humano. O qual possui uma característica em seu formato, onde a partir classificação das faces notou-se a existência de praticamente apenas uma única superfície conexa. A existência de uma única superfície é muito interessante, pois, futuramente é provável fazer uma análise da sua forma, em resolução aumentada, através dessa superfície, tal como descrito por [2] em resolução normal.

Referências

- [1] G. Bertrand and M. Couplie. New 3d parallel thinning algorithms based on critical kernels. In A. Kuba, K. Palágyi, and L. Nyúl, editors, *DGCI*, LNCS. Springer Verlag, 2006.
- [2] S. Bouix, J. C. Pruessner, D. L. Collins, and K. Siddiqi. Hippocampal shape analysis using medial surfaces. *Neuroimage*, 25:1077–89, 2005.
- [3] M. Couplie, A. V. Saúde, and G. Bertrand. Euclidean homotopic skeleton based on critical kernels. In *SIBGRAPI*, pages 307–314. IEEE CS press, 2006.
- [4] G. Malandain, G. Bertrand, and N. Ayache. Topological segmentation of discrete surfaces. *International Journal of Computer Vision*, 10:2, 183-197 (1993) © 1993 Kluwer Academic Publishers, Manufactured in The Netherlands., 1993.

SketchSim: Interactive Simulation of Ink Brushstrokes

Nicole Sultanum and Silvio B. Melo (orientador)
Centro de Informática - UFPE

Abstract

There is an increasing demand for new computational solutions in art and design, for digital illustrations, games, animation, among others. In order to address this need, several ideas have been developed in the artistic field, and one of them is concerned with the simulation of real artistic instruments, such as pencil, watercolor, etc. The simulation of ink and brushes, one of these instruments, is the object of study of this paper. Several techniques are discussed and implemented in an interactive artistic tool, called SketchSim. Artists have used this tool as part of a brief qualitative analysis, which is also discussed in this paper.

1. Introduction

The computational simulation of artistic instruments (such as pencil, charcoal, pastel, watercolor, among others) is a steadily growing field in NPR [2] (or Non-Photorealistic Rendering), in face of the current needs for more sophisticated and intuitive tools for the creation of drawings. In digital illustration, the simulation of such instruments is sought by artists who wish to ease or improve their creative experience through computational solutions that offer new features, such as making slight adjustments, undoing strokes or trying out numerous colors. These functionalities are among those which, most of the times, may only be seized through digital means.

Nevertheless, the effective digital simulation of a real artistic tool is usually followed by great challenges. Here, we define “effective” solution as one which goes beyond the proper simulation of the real tool, and also:

- (1) Provides intuitive means for the creation of artistic strokes;
- (2) Offers resources for later modifications on the strokes, as part of the complementary experience the computational environment should provide;
- (3) Renders fast enough for interactive use.



Figure 1. “Cat”, illustration produced on SketchSim¹

According to this scenario, this paper joins several techniques for the simulation and post-edition of ink brushstrokes, applied in the implementation of an interactive drawing tool called SketchSim.

1.1. Related Work

Many strategies have been proposed, concerning the simulation of artistic brushstrokes. Strassmann [6] presents a physically based approach for rendering strokes in *sumi-e* style, which produces realistic results, but requires too much user intervention and is too slow for interactive use. Skeletal Strokes [3] are also inappropriate for our purposes since there is a constant dependency of an image in order to reproduce different styles and is also too computationally intensive (since it is based on texture mapping strategies), not mentioning its weaknesses on resizing operations (as pointed by Su et. al. [7]). Su et. al. [7] present techniques for the creation of analytical strokes with variable width, having the disadvantage of requiring the explicit specification of control knots. Pudet [4] has also created analytical strokes with variable width, which are mapped to pressure levels acquired from a tablet device. These works usually spend little or no effort on fulfilling all of the additional requirements mentioned in the Introduction (intuitiveness, flexibility and efficiency).

¹ By Zózimo Neto.

2. Simulation of ink brushstrokes

Our stroke rendering process was heavily inspired on the work of Schneider [5] and Pudet [4]. A tablet is used as primary input device, since it provides pressure data, and a more intuitive interaction.

This process is composed of two main steps. The first one, the curve fitting step, creates a vector description for the user trajectory, specified through the tablet. In *SketchSim*, such as in [5], piecewise cubic Bézier curves with G^1 continuity were chosen to analytically represent the trajectories. The second one, the border fitting step, is responsible for simulating the appearance of an artistic stroke surrounding the vector trajectory which is resulted from the first step. Both of these steps are described on the next sections.

2.1. Curve Fitting

This step concerns the creation of an analytical description for the user trajectory. Initially, this trajectory is represented by a sequence of two-dimensional points (plus the corresponding pressure value, which is used in the border fitting step) provided by the tablet device, each of them captured in regular intervals of 10ms. This sequence goes through a few pre-processing steps (inspired by Schneider [5]), and is subsequently submitted to a least squares algorithm.

The trajectory points firstly undergo a pre-reduction step, for the high sampling frequency of the tablet device usually produces redundant point sets: points which are closer than 5 pixels are ignored. Discontinuities are also checked, represented by pointy corners, which are detected by the existence of angles sharper than 140° in the trajectory. Schneider also proposes another two pre-processing steps, which were eventually discarded. One of them, the noise removal step, was contributing negatively for the curve fitting process (Pudet [4] has also observed this phenomenon). The other step, the linear splines reduction, was also not used: it was proposed as an ultimate reduction of the sampling set, but the author himself mentioned that it could be discarded if the rendering process were to work fast enough without it (which has happened, indeed).

After pre-processing, the curve fitting process may start. Schneider [5] describes it in detail. In short, it consists of several iterations of (1) fitting of a curve and (2) evaluation of the resulting curve. The stage (2) measures the quality of the fit of the curve produced in (1), by checking the distance from the sampling points to the curve. The Newton-Raphson method is used in order to find the point on the curve which lies closest to each sampling point. If the fit doesn't reach an

acceptable closeness after a predefined number of iterations, then the points sequence is subdivided on the point of greatest error, and the curve fitting procedure restarts for each of the new sequences. It results in the creation of several cubics to represent a trajectory, as illustrated by the rightmost image on Figure 2.



Figure 2. Steps of the curve fitting algorithm

2.2. Border Fitting

The border fitting step starts right after the curve fitting step. The technique described in this section is based in the work of Pudet[4], and may be considered as a simpler version of his border fitting method, specialized in circular brushes.

The tablet provides, for each sampling point, a value corresponding to the pressure applied on that location. The curve fitting step also computes the corresponding parametric value t_i of each sampling point in the trajectory. With these data at hand, it is possible to associate pressure data directly into the analytical curve.

Take $press_i$ as the corresponding normalized pressure value of the sampling point d_i , and el as an elasticity value which limits the maximum width a stroke may have. The normalized tangent for each parametric position of samples d_i (let's say, $Q(t_i)$) is computed through the De Casteljau algorithm. Then, in the direction perpendicular to the tangent at $Q(t_i)$, two points are located in this trail. These two points have distance to $Q(t_i)$ equal to $el \times press_i$, and represent points on each of the right and left borders. An example is illustrated in Figure 3.

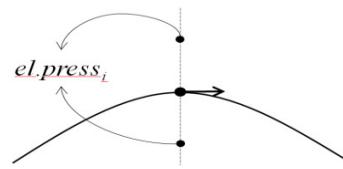


Figure 3. Border Points

This procedure is repeated for all samples d_i , resulting in two sequences: the left and right border points. Finally, these sequences are vectorized by the same curve fitting algorithm described in section 2.1.

The final result consists of two independent piecewise Bézier cubics, one for each border side.

Figure 4 shows, in the middle, the border points generated after the analytical trajectory on the left is processed. On the right side, one can see the final stroke, after applying curve fitting to the borders.



Figure 4. Border Fitting steps

3. Stroke Edition

Bartels and Beatty [1] presented a simple technique for modifying Bézier curves based on the displacement of any point in the curve, only adjusting its control points. This strategy is very convenient since it allows the implementation of drag-and-drop-based adjustments. The ability to choose which control points should be moved is another advantageous characteristic of this technique. Meanwhile, only the adjustment of isolated Bézier curves (and B-splines, which are not relevant to us) were analyzed in [1]. In this work a novel mechanism was required; one that could maintain G^1 continuity between adjacent segments after adjustment of any cubic in the analytic trajectory.

Nevertheless, before adjusting the trajectory, it is required to detect which of its cubics must be modified, and also to identify which point of this curve should be displaced (more specifically, the parametric t value which represents a point in a Bézier curve), taking the user specified start point $p_s = (x, y)$ as the sole input (the beginning of the drag-and-drop operation). The problem is then reduced to the computation of the distance between a point p_s and a Bézier curve. We used a technique based on subdivision, which divides the curve into smaller pieces until they are close enough to line segments. When such condition is reached, the projection of the point p is computed on the line segment defined by the endpoints of this sub curve. If the projection is placed between the endpoints, an interpolation is made between the endpoint parameter values (which are known) to estimate the t parameter of the projection. Several candidate points on the curve may be found through this process. We choose the one which has the smallest squared distance from p_s .

After determining t , it is possible to perform the displacement of the control points. Let's define that an arbitrary trajectory is composed of n adjacent cubics c_0 to c_{n-1} . Supposing that c_k , $0 < k < n-1$, was identified as the cubic to be displaced, then its control points $b_{0,k}$,

$b_{1,k}$, $b_{2,k}$ and $b_{3,k}$ will be modified to the new points $\hat{b}_{0,k}$, $\hat{b}_{1,k}$, $\hat{b}_{2,k}$ and $\hat{b}_{3,k}$, such that the corresponding curve \hat{c}_k passes through the new point p_e (the endpoint of the drag-and-drop operation). After that, the curves c_{k-1} and c_{k+1} must be readjusted, in order to assure that the points $\hat{b}_{2,k}$, $\hat{b}_{3,k}$ (which is equal to $\hat{b}_{0,k+1}$) and $\hat{b}_{1,k+1}$ are collinear (as shown in Figure 5). The same condition must be verified on the points $\hat{b}_{1,k}$, $\hat{b}_{0,k}$ (equal to $\hat{b}_{3,k-1}$) and $\hat{b}_{2,k-1}$.

Let's rename the points $b_{2,k}$, $b_{3,k}$ e $b_{1,k+1}$ to a , b and c , as illustrated on Figure 5. After the adjustment of the curve c_k , a and b are modified to \hat{a} and \hat{b} , respectively. The point \hat{c} is defined as a point lying on the line defined by \hat{a} and \hat{b} which keeps the same proportion between the segments ab/bc (composed by the earlier points a , b and c) and $\hat{a}\hat{b}/\hat{b}\hat{c}$.

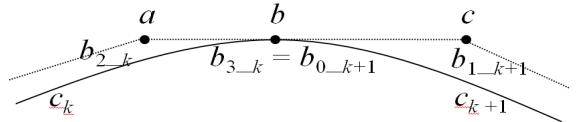


Figure 5. The control points of two adjacent curves in a trajectory

Not always are all the four control points adjusted on each modification. As a matter of fact, for the first cubic on a trajectory, its control point $b_{0,0}$ is locked, and only $b_{1,0}$, $b_{2,0}$, and $b_{3,0}$ are displaced. This behavior was adopted since it was taken as more intuitive and predictable.

Nonetheless, if the user tries to displace a point close enough to $b_{0,0}$, it is expected that one wishes to modify the endpoint of the curve, and then $b_{0,0}$ is also moved (as suggested by [1]).

This whole adjustment process is applied to the *core trajectory* of a stroke (the zero-width curves generated after the curve fitting step). After this edition process, the borders must be recalculated for the new trajectory, as described in the Section 2.2.

4. Results and Discussion

As discussed in the introduction of this paper, this work was focused on the creation of an ink brushstrokes simulator which, beyond properly renderizing the strokes, could also satisfy other requirements: provide efficiency, complementary resources and intuitive means of user interaction.

The form of interaction is quite simple and straightforward, since the artist may work with a tablet device almost as if it was a regular pen (which is a very familiar tool). During the drawing of a stroke, an approximate outline of the final result is presented to

the user. When it is finished, the trajectory is computed and fully rendered. Experiments have shown that the whole process operates quickly enough, allowing its use on interactive systems.

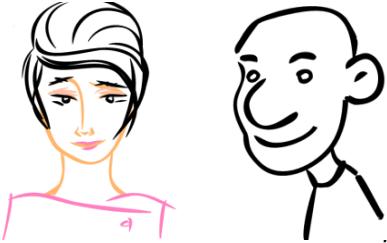


Figure 6. Untitled Illustrations²

In order to evaluate the user interaction and the quality of the strokes, *SketchSim* was used and analyzed by eight end users, including designers and digital artists. Figures 1, 6 and 7 represent some of the resulting illustrations. They evaluated positively the tool, particularly emphasizing the intuitiveness of the stroke creation process in comparison to the quality of the strokes created. They also observed that the generated curves were coherent and adherent to what they were trying to draw.



Figure 7. "Airman"³

This study has also encountered some problems concerning the stroke rendering process, consequence of the instability of the Newton-Raphson method. This technique, used in the curve fitting algorithm, eventually computes inadequate values which result in anomalous strokes.



Figure 8. Example of an inappropriately local adjustment

A small issue was also observed on the stroke edition technique, due to the fact that the maintenance of continuity considers only the immediately adjacent

² By Frederico de Melo (left) and Antunes Neto (right).

³ By Fradique Filho.

segments. When the cubics generated by the curve fitting algorithm are too short, the edition effects are also reflected in a very small area of the stroke, resulting in an uneven appearance, as illustrated in Figure 8.

5. Conclusions and Future Work

An artistic ink brushstroke simulator was developed, which uses pressure data provided by a tablet device to represent width variations on a stroke. The strokes are represented by analytical curves which provide flexibility on further readjustments while maintaining resolution quality. A stroke edition technique was developed, based on intuitive 'drag and drop' operations. The simulator was also evaluated by artists, validating its artistic potential.

One possible future work that could increase this potential is the simulation of ink dilution effects (such as in [7]). Another idea would be to explore more advanced types of curve modification beyond position readjustment. Width edition along the strokes, for example, would be quite helpful for the artists. Another possible example of stroke edition is the redraw, as proposed by Schneider [5] for zero width curves, but in this case considering the continuity aspects of the borders on the endpoints of each edited portion of a curve.

6. References

- [1] R. H. Bartels and J. C. Beatty, "A Technique for the Direct Manipulation of Spline Curves". *Proceedings of Graphics Interface '89*, p.33-39, 1989.
- [2] B. Gooch and A. Gooch, *Non-Photorealistic Rendering*, A. K. Peters, 2001.
- [3] S. C. Hsu and I. H. H. Lee, "Drawing and animation using skeletal strokes". *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM Press, pp. 109-118, 1994.
- [4] T. Pudet, "Real Time Fitting of Hand-Sketched Pressure Brushstrokes". *Computer Graphics Forum*, 13(3), pp. 205-220, 1994.
- [5] P. J. Schneider, *Phoenix: An interactive curve design system based on the automatic fitting of hand-sketched curves*, Master's Thesis, University of Washington, 1988.
- [6] S. Strassmann, "Hairy Brushes", *Computer Graphics (SIGGRAPH '86 Proceedings)*, 20(4), pp. 225-232, 1986.
- [7] S.L Su et. al., "Simulating Artistic Brushstrokes Using Interval Splines". *The 5th International Conference on Computer Graphics and Imaging*, pp. 85—90, 2002.

Uso de Saliências do Contorno via Esqueletização para Caracterização de Formas

Glauco V. Pedrosa¹ (Bolsista PIBIC/CNPq), Marcos A. Batista², Henrique C. Fernandes¹,
Celia A. Z. Barcelos¹ (Orientadora)

¹ Universidade Federal de Uberlândia ² Universidade Federal de Goiás

E-mail: glaucovitor@comp.ufu.br, marcos@catalao.ufg.br, celiazb@ufu.br

Resumo

Neste trabalho, mostraremos o descritor saliências do contorno para a caracterização de formas juntamente com o algoritmo de casamento proposto para medir a distância entre duas formas caracterizadas pelos seus pontos de saliências. Alguns experimentos são mostrados e discutidos para demonstrar o potencial do descritor na recuperação de formas.

1. Introdução

Inúmeras áreas do conhecimento humano vêm se apoiando cada vez mais no uso de imagens e, como consequência, surge a necessidade de armazenar, manter e consultar grandes bases de dados de imagens eficientemente. Assim, nasceram os CBIR's (CBIR - Context Based Imagem Retrieval), que são sistemas que realizam a recuperação de imagens.

A recuperação de imagens pode ser feita baseada principalmente em três aspectos: cor, forma e textura, sendo que, em um sistema de recuperação, se pode utilizar apenas um ou mesmo uma combinação desses três elementos. Em algumas situações, a forma é o aspecto mais importante, sendo que invariância quanto a transformações é imprescindível.

Uma maneira eficiente de representar uma forma é através de suas saliências. Elas são definidas como os pontos de maior curvatura ao longo do contorno, por isso capturam a essência de uma forma, uma vez que qualquer parte sobressalente ou mesmo reentrâncias são consideradas saliências, e são essas as características que diferenciam duas formas.

Neste artigo abordaremos a recuperação de imagens baseada em forma utilizando as saliências de contorno como descritor. As saliências serão obtidas a partir do esqueleto como em [2] utilizando uma técnica denominada Transformada Imagem Floresta. Juntamente apresentaremos um algoritmo de casamento

originalmente proposto em [1]. Por fim, mostraremos alguns resultados experimentais da implementação de um CBIR baseado em formas usando o descritor.

2. IFT – Image Foresting Transform

Uma forma eficiente de se encontrar as saliências é através da técnica denominada Transformada Imagem Floresta (IFT- Image Foresting Transform), bastante explorada em [1; 2; 3; 4], que utiliza grafos para a representação de uma imagem através de seus relacionamentos de conectividade: os nodos são pixels e os arcos são relações de adjacência entre os pixels. A IFT é uma transformada projetada para operadores de processamento de imagem que gera uma representação em multiescala.

Cada ponto do contorno é tomado como uma semente e a partir desta um caminho de custo mínimo é determinado. Cada caminho constitui uma árvore e a união delas constitui uma floresta cobrindo a imagem inteira. A IFT gera duas diferentes representações da forma: contornos multiescala por diliação exata e esqueletos multiescala por propagação de rótulos.

O algoritmo da IFT utiliza um conjunto de sementes S formado pelos pixels do contorno. Cada elemento de S possui um rótulo diferente e o objetivo é propagá-los por toda a imagem. Para isso, deve-se percorrer todos os pixels da imagem e encontrar o pixel semente correspondente, em S , com menor custo. O custo é calculado de acordo com a distância euclidiana e caso seja mínimo entre um pixel semente (pertencente ao contorno) e um pixel qualquer, o rótulo da semente é atribuído a este pixel. Se o pixel já pertencer ao conjunto S o custo dele é 0.

No final do algoritmo, teremos um mapa de raízes R e um mapa de custo C para todos os pixels da imagem.

A figura 1-c mostra o resultado da IFT aplicada à imagem 1-a. A propagação dos rótulos das sementes de S resulta em uma imagem totalmente rotulada.

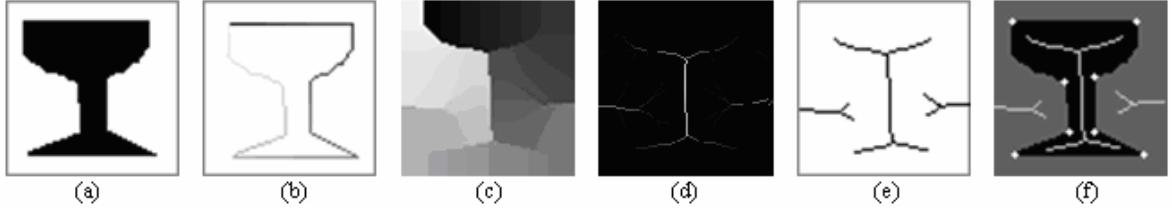


Figura 1: Saliências através da IFT. (a) Forma original. (b) Contorno rotulado. (c) Imagem rotulada. (d) Imagem diferença. (e) Threshold aplicado à imagem diferença. (f) Pontos claros são saliências obtidas da relação entre a extremidade do esqueleto e o contorno.

3. O processo de rotulação

Para a obtenção do conjunto de sementes S , necessários para a aplicação da IFT, é aplicado o algoritmo de rotulação, o qual encontra as sementes e atribui rótulos incrementais às mesmas começando de 1. Assim, os rótulos variam no intervalo de 1, 2, ..., N onde N é o número de pixels do contorno. A estratégia usada para o algoritmo de rotulação é visitar cada pixel do contorno uma única vez caminhando no sentido anti-horário (ou horário) atribuindo um rótulo diferente a cada um.

Na figura 1-b pode ser visto o resultado da rotulação dos pixels do contorno da figura 1-a.

4. Esqueletos

Inúmeras são as aplicações de esqueletos devido à sua capacidade de representar formas de maneira compacta, dentre essas aplicações tem-se a obtenção das saliências visto que, toda extremidade de esqueleto interno, corresponde a uma saliência convexa, e extremidades do esqueleto externo correspondem a saliências côncavas.

Esqueletos multiescala são facilmente obtidos da IFT, já que desta resulta um mapa de rótulos L que possibilita separar os pixels em regiões discretas de Voronoi. O esqueleto é extraído a partir das divisões destas regiões, sendo que a obtenção da imagem diferença D é a parte responsável por tal etapa, seu cálculo é feito pela seguinte equação:

$$D(p) = \max_{\forall q \in A_4(p)} \{\min(\delta(p, q), N - \delta(p, q))\} \quad (1)$$

onde $A_4(p)$ são os pixels na vizinhança-de-quatro relativa ao pixel p , $\delta(p, q) = L(q) - L(p)$, e $L(q)$ é o rótulo do pixel q . Dessa equação pode se observar que valores altos são gerados quando os valores dos pixels em L são muito diferentes, caso contrário valores baixos serão obtidos. Ao representar a imagem diferença,

figura 1-d, pode-se perceber linhas mais claras próximas ao centro da forma em que se tem uma forte diferença entre os pixels, ou seja, trata-se de linhas significativas do esqueleto, tais linhas escurecem à medida que se aproximam das extremidades pois nesses locais a diferença entre os pixels não é tão acentuada como no centro da forma. Portanto, quanto maior a diferença de um pixel para seus vizinhos, maior o valor atribuído a esse pixel e maiores são as chances de que ele pertença ao esqueleto.

A imagem diferença possui inúmeros detalhes sendo que a maioria não pode ser vista através do sistema visual humano, estes são os detalhes menos relevantes que geram linhas muito escuras e imperceptíveis e muitas vezes não devem estar presentes no esqueleto. A fim de obter um esqueleto que represente a forma convenientemente, aplica-se um threshold na imagem diferença, ou seja, um limiar que separa os pontos com tons mais claros (os mais relevantes) dos mais escuros (os menos relevantes).

Uma estimativa para tal valor, é encontrada em [3], que sugere 5% de N , sendo que N é o número de pixels do contorno e por consequência o número de regiões discretas de Voronoi [5]. Esta estratégia se fundamenta no fato de que a máxima diferença que pode ser obtida é $N/2$ e que, no entanto, não se pode obter diferenças tão grandes já que uma vizinhança de quatro está sendo analisada e os pixels vizinhos tendem a ter valores próximos devido à rotulação incremental.

A figura 1-e mostra o resultado do threshold aplicado à imagem 1-d.

5. Geração do conjunto de características

Os valores contidos na imagem diferença, obtida pela equação 1, podem ser interpretados como sendo o comprimento máximo do segmento mais curto entre duas raízes equidistantes [3], segmento dab , figura 2. Com isso, é necessário apenas fazer a relação entre a extremidade do esqueleto e o meio deste segmento, já que esta é a saliência. Desta forma, partindo da extremidade do esqueleto c , deve-se encontrar a raiz

deste pixel através do mapa raiz R (obtido pela IFT) e em seguida percorrer a distância $\overline{dab}/2$ no sentido horário (ou anti-horário). Com isso a posição da saliência torna-se conhecida.

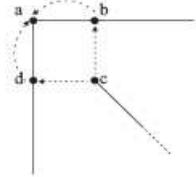


Figura 2: Relação entre o esqueleto e a saliência.

Obtida a posição da saliência, é necessário atribuir um valor para a mesma. Aqui novamente a IFT é de grande importância: através do mapa de raízes R e o mapa de custo C , obtidos pela IFT, é possível obter o tamanho da área de influência de cada ponto do contorno através da simples construção de um histograma. O valor da saliência é tomado como sendo a área de influência interna mais a área de influência externa da saliência.

Falta apenas determinar se a saliência é côncava ou convexa. No primeiro caso o valor da saliência torna-se negativo e no segundo permanece positivo. Se a extremidade é de esqueleto interno, a saliência é convexa, caso contrário é côncava.

Para tornar o conjunto de características mais eficiente a posição a ser utilizada deve ser a posição relativa da saliência com relação ao contorno: escolhe-se um ponto arbitrário como sendo o ponto inicial, com isso cada ponto tem uma posição. Em seguida, divide-se o valor da posição pelo total de pontos no contorno. Essa última medida visa garantir que haja normalização quando mudanças de escala são realizadas.

Portanto, uma saliência b de uma forma é representada pelo par (u_b, s_b) , onde $u_b \in [0,1]$ é a posição relativa e $s_b \in [0,1]$ é o tamanho da área de influência.

6. Algoritmo de Casamento

O objetivo do algoritmo de casamento é medir a distância entre duas formas A e B . Essas formas são, respectivamente, representadas pelos vetores:

$$F_A = \{(u_{A1}, s_{A1}), \dots, (u_{An}, s_{An})\}$$

$$F_B = \{(u_{B1}, s_{B1}), \dots, (u_{Bm}, s_{Bm})\}$$

onde (u_{Ai}, s_{Ai}) , representa a i -ésima saliência da forma A , u_{Ai} é a posição relativa e s_{Ai} é o valor que representa o tamanho da área de influência.

O número de saliências n de A pode ser diferente do número de saliências m de B . O casamento entre as formas A e B é definido pelo algoritmo:

1. Crie $F_A' = \{(u_{A1}', s_{A1}'), \dots, (u_{An}', s_{An}')\}$ e $F_B' = \{(u_{B1}', s_{B1}'), \dots, (u_{Bm}', s_{Bm}')\}$, ordenando F_A e F_B por ordem decrescente da área de influência.
2. Cria a lista L contendo pares de pontos candidatos a casamento de F_A' e F_B' . Um par $\left(\left(u_{Ai}', s_{Ai}'\right), \left(u_{Bj}', s_{Bj}'\right)\right)$ está em L se $|s_{Ai}' - s_{Bj}'| \leq 0.2 * s_{A1}'$. Um par $\left(\left(u_{Bj}', s_{Bj}'\right), \left(u_{Ai}', s_{Ai}'\right)\right)$ está em L se $|s_{Bj}' - s_{Ai}'| \leq 0.2 * s_{B1}'$.
3. Para cada par de candidatos a casamento na forma $P_{ij} = \left(\left(u_{Ai}', s_{Ai}'\right), \left(u_{Bj}', s_{Bj}'\right)\right)$ em L , encontre o parâmetro de deslocamento α tal que $\alpha = u_{Ai}' - u_{Bj}'$. Desloque os pontos de saliências de F_A por α , obtendo $F_A'' = \{(u_{A1}'', s_{A1}''), \dots, (u_{An}'', s_{An}'')\}$
4. A distância d entre F_A'' e F_B é dada por:

$$d = \sum_{k=1}^{\min\{n,m\}} d_k$$
onde

$$d_k = \begin{cases} \sqrt{(u_{Ak}'' - u_{Bk})^2 + (s_{Ak}'' - s_{Bk})^2}, & \text{se } |u_{Ak}'' - u_{Bk}| \leq 0.2 \\ s_{Ak}'' + s_{Bk} & \text{caso contrário} \end{cases}$$
Finalmente, se $n \neq m$, uma altura s dos pontos não casados é adicionada à soma d .
5. Repita os passos 3 e 4 considerando outros pares de candidatos em L
6. Selecione a menor distância d como sendo a distância entre F_A e F_B .

Query	Ranking									
	1	2	3	4	5	6	7	8	9	10
A										
B										
C										

Tabela 1. Resultado das 10 primeiras posições usando as queries A, B, C

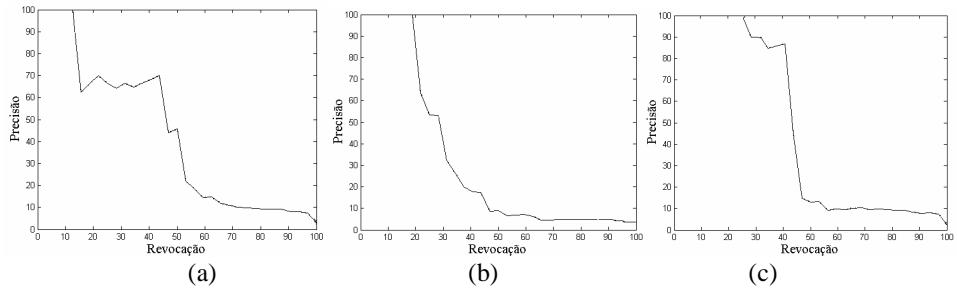


Figura 3 - Gráfico Precisão x Revocação da queries da tabela 1: (a) Query A (b) Query B (c) Query C

7. Resultados Experimentais

Implementamos um CBIR baseado em formas, a fim de testar a construção e o comportamento do descriptor num ambiente de recuperação de imagens.

A base de dados é formada por 40 tipos diferentes de formas, e cada uma possui 32 variações quanto à escala e rotação. A imagem da forma juntamente com suas variações é considerada uma classe e as imagens relevantes de uma imagem consulta é a classe a qual essa imagem consulta pertence.

Nas consultas uma imagem exemplo é fornecida, suas características são extraídas, a busca é efetuada e um ranking é construído. A tabela 1 mostra o ranking das 10 primeiras posições de 3 imagens, usadas como consulta. Na primeira consulta, podemos observar que as 6 primeiras posições são relevantes. Na segunda, apenas as posições 1 a 4, 8, 9, 10 são relevantes. Na terceira, apenas a posição 9 não é relevante.

8. Conclusão

Este trabalho apresentou o descriptor saliências do contorno, usado para caracterizar uma forma. Implementamos um CBIR baseado em formas e utilizamos esse descriptor para analisar seus resultados na recuperação de formas. Os experimentos mostram o potencial da técnica.

9. Referências

- [1] R. da S. Torres, E. M. Picado, A. X. Falcão, and L. da F. Costa, “Effective Image Retrieval by Shape Saliences”. SIBGRAPI, 2003, 166-174, 12-15.
- [2] Falcão, A. X., Costa, L. F., and Cunha, B. S. “Multiscale Skeletons by Image Foresting Transform and its applications to neuromorphometry”. *Pattern Recognition*, 35, pp. 1571–1582, 2002.
- [3] R. da S. Torres, A. X. Falcão, and L. da F. Costa. “A graph-based approach for multiscale shape analysis”. *Pattern Recognition*, 37(6), pp. 1163–1174, 2004.
- [4] Torres, R. S., and Falcão, A. X. “Contour Salience Descriptors for Effective Image Retrieval and Analysis”. *Image and Vision Computing journal*, pp. 1–11, 2006.
- [5] Hoff III, K. E., Culver, T., Keyser, J., Lin, M. e Manosha, D.. “Fast computation of generalized Voronoi diagrams using graphics hardware”. Proceedings of ACM SIGGRAPH using graphics hardware, 2000.
- [6] Costa, L. da F., Campos, A. G., Manoel, E. T. M.: “An Integrated Approach to Shape Analysis: Results e Perspectives”. International Conference on Quality Control by Artificial Vision, Le Cresot, France, (2001) pp. 23-24.
- [7] Costa, L. da F., Estrozi, L. F.: “Multiresolution Shape Representation without Border Shifting”. *Electronic Letters*, Vol.35 (1999) pp. 1829-1830.
- [8] Arica, N., Vural, F. T. Y.: “A Perceptual Shape Descriptor”. In International Conference on Pattern Recognition, (2002) pp. 375-378.

Técnicas Simples para Visualização Foto-realista de Terrenos em Tempo Real Usando Placas Gráficas Programáveis

Ricardo Gomes Leal Costa

Waldemar Celes (orientador)

Tecgraf - Departamento de Informática, PUC-Rio

{rcosta, celes}@inf.puc-rio.br

Resumo

Este trabalho consiste em explorar técnicas simples de programação em placas gráficas para gerar imagens de terrenos virtuais com boa qualidade. Implementamos vários efeitos de iluminação que, somados ao mapeamento procedural de camadas de textura em função da altitude, resultam numa aparência mais natural. Empregamos o algoritmo Variance Shadow Map para geração de sombras suaves em terrenos. Apresentamos ainda uma solução simples e eficiente para simulação de neblina e a variação da cor do céu em relação à hora do dia. Os resultados obtidos foram positivos, alcançando uma aparência realista com ótimo desempenho.

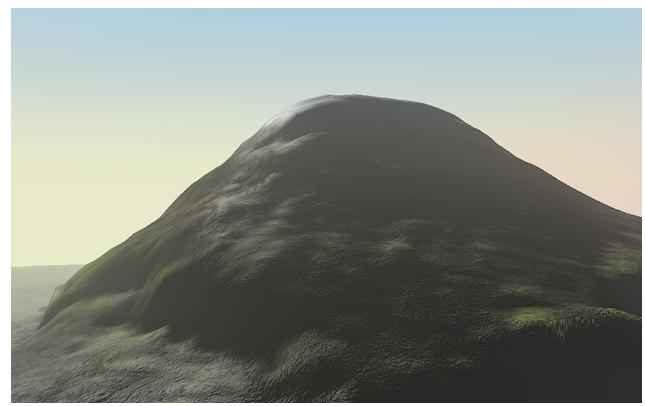


Figura 1. Terreno ao amanhecer.

1. Introdução

A visualização de terrenos é utilizada em diversas áreas da computação como jogos, simuladores de vôo, mapeamento de planetas e simulações científicas. Em muitos casos, deseja-se obter uma aparência foto-realista dada pelas características geométricas (relevo e rugosidade da superfície), materiais (cor e reflexão luminosa de cada tipo de solo) e ambientais (efeitos climáticos e atmosféricos). A simulação destes efeitos geralmente requer algoritmos computacionalmente caros que nem sempre são adequados para visualização em tempo real.

O objetivo deste trabalho é apresentar uma solução simples para esses problemas através de programação em placa gráfica. Todos os cálculos de geometria e iluminação são feitos na unidade de processamento gráfico (GPU) a partir de parâmetros definidos na unidade de processamento central (CPU). Com isso, espera-se obter um bom desempenho mesmo com a grande quantidade de cálculos necessária.

Os vértices do terreno, que formam uma malha inicialmente plana, são deslocados por um mapa de elevação. Em seguida, o programa de fragmentos mapeia a textura correspondente à altitude daquele ponto e calcula a iluminação di-

fusa e especular, aplicando também mapas de rugosidade e de brilho. As sombras são geradas pelo algoritmo *Variance Shadow Map* [3], que apresenta diversas vantagens quando usado em terrenos. Finalmente, o terreno recebe um efeito de névoa seca. O Sol e a cor do céu também são desenhados proceduralmente pelo programa de fragmentos com base na hora do dia.

2. Implementação das técnicas

2.1. Geometria

O relevo do terreno é gerado a partir de uma malha de triângulos plana com $y = 0$ para todos os vértices. A quantidade de vértices não varia ao longo do tempo, o que torna vantajoso seu armazenamento em memória de vídeo por meio de um *vertex buffer object*. O relevo do terreno é descrito por uma textura chamada *mapa de elevação*. Esta textura é acessada pelo programa de vértices e funciona como um *mapa de deslocamento* para a superfície do terreno, ou seja, a nova coordenada y dos vértices é proporcional ao va-

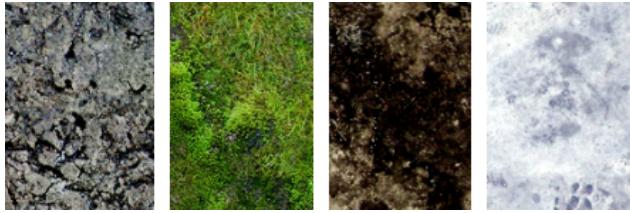


Figura 2. Texturas com os tipos de solo.

lor dos texels correspondentes.

Em seguida, é necessário calcular as normais dos vértices. Para isso, calculamos a elevação dos vértices vizinhos, acessando seus respectivos texels no mapa de elevação. São necessários apenas dois vértices vizinhos para formarmos uma base de três vértices, portanto um acesso é feito na coordenada de textura $(s + d, t)$ e outro em $(s, t + d)$, onde d é a distância entre dois texels em coordenadas de textura. Em posse da posição dos vértices vizinhos, o produto vetorial dos vetores formados por esses pontos resulta na normal daquele vértice, que é passada para o programa de fragmentos como coordenada de textura para cálculo da iluminação.

2.2. Aparência

Em geral, um terreno possui camadas de diferentes materiais que variam com a altitude. Neste trabalho, consideramos que o terreno apresenta quatro tipos de solo de acordo com sua elevação: arenoso, gramado, rochoso e com neve (Fig. 2). Estas quatro texturas são passadas em conjunto ao programa de fragmentos através de um *array de texturas* [5]. A vantagem desta técnica em relação ao uso de quatro texturas separadas é que apenas uma unidade de processamento de texturas da GPU é utilizada. Também é mais conveniente que uma textura volumétrica, pois os filtros de redução e magnificação são aplicados apenas nas coordenadas s e t , dando liberdade para fazermos a interpolação entre camadas da maneira desejada, proceduralmente.

No programa de fragmentos, cada fragmento recebe a altitude dos vértices interpolada linearmente pelo rasterizador. Com base nesta altitude, uma das quatro texturas é selecionada e seu texel contribui para a cor difusa do fragmento. Nas interseções entre camadas, as duas texturas são interpoladas. A Tabela 1 mostra um exemplo de como cada textura pode ser escolhida em função da altitude do fragmento normalizada para o intervalo $[0, 1]$.

Técnicas procedurais são utilizadas para tornar as transições de camadas menos artificiais. Primeiro, uma textura volumétrica contendo *Perlin Noise* [6] é acessada com as mesmas coordenadas de textura utilizadas para o mapeamento dos tipos de solo, e o valor resultante é somado aos

Altitude	Textura
0.00 - 0.25	Areia
0.20 - 0.45	Grama
0.40 - 0.85	Rocha
0.80 - 1.00	Neve

Tabela 1. Camadas de textura.

limites de altitude entre camadas da Tabela 1, deixando as transições menos lineares e homogêneas. Inicialmente, tentamos somar a perturbação diretamente à altitude do ponto, mas isto resultou numa aparência granulosa para superfícies planas situadas nas transições, o que não parecia muito natural. Em seguida, a função *smoothstep* calcula a interpolação entre as camadas. Ela cria uma transição não linear entre dois valores, o que também deixa as transições de camadas menos perceptíveis.

2.3. Iluminação

A iluminação difusa, especular e ambiente é calculada por fragmento pelo modelo de reflexão de *Phong* [7]. É possível definir coeficientes distintos de reflexão difusa, especular, ambiente, emissiva e fator de brilho para cada camada do terreno. Assim, a camada de gelo pode ter reflexão especular mais forte que a camada de areia, por exemplo. Nas áreas de transição de camadas, estes coeficientes são interpolados do mesmo modo que as texturas.

2.3.1. Mapeamento de rugosidade Para dar aparência rugosa ao terreno, não podemos simplesmente alterar a geometria, pois a quantidade de triângulos necessária para representar essas pequenas deformações seria inviável do ponto de vista do desempenho. Para contornar este problema, a técnica de *mapeamento de rugosidade* é utilizada. Ela cria perturbações nas normais da superfície para que a iluminação dê à superfície uma aparência rugosa. Para isso, um *mapa de normais* contém em seus canais (R, G, B) as coordenadas (x, y, z) das normais perturbadas de uma superfície plana. Assim como as texturas do terreno, o mapa de normais também é passado ao programa de fragmentos por um array de texturas, já que cada camada pode apresentar rugosidade diferente. Porém, como a superfície do terreno é irregular, não podemos aplicar diretamente essas normais. Em vez disso, os parâmetros para o cálculo da iluminação são transformados para o *espaço tangente*. Ele é obtido através de uma base ortonormal contendo os vetores normal, binormal e tangente de cada ponto, previamente calculados no programa de vértices. Dessa forma, as perturbações das normais tornam-se relativas às normais originais do terreno.

Para que a iluminação com mapeamento de rugosidade pareça correta, é necessário considerar não apenas a nor-

mal perturbada como também a original no cálculo de iluminação difusa e especular. Existem casos em que a normal original está voltada contra a luz, mas a perturbada não, e vice-versa. Quando isso acontece, surge rugosidade em áreas não iluminadas, ou áreas iluminadas ficam sem rugosidade. A solução mais simples para este problema é iluminar apenas quando ambas as normais não estão contra a luz. Porém, esta definição implica numa transição abrupta entre regiões iluminadas e regiões em sombra (Fig. 3a). Por conta disso, optamos por interpolar o fator de auto-sombreamento de ambas as normais suavemente com uso da função *smoothstep*, resultando numa aparência mais natural (Fig. 3b).

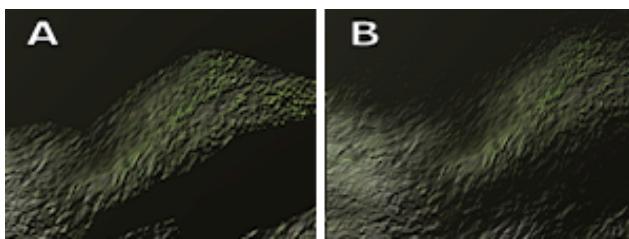


Figura 3. Correção da iluminação com mapeamento de rugosidade.

2.3.2. Mapeamento de brilho Além da rugosidade, algumas regiões da superfície do terreno podem apresentar mais ou menos reflexão especular. Por exemplo, uma rachadura na rocha não apresentaria nenhuma reflexão especular, enquanto que pequenos cristais de silicato teriam uma reflexão especular forte. Para simular esse efeito, uma textura especial em tons de cinza mapeia as diferentes intensidades de brilho especular, funcionando como um coeficiente para o valor especular resultante daquele fragmento.

2.3.3. Geração de sombras Para que as montanhas do terreno gerem sombras ao serem iluminadas pelo sol, precisamos implementar uma técnica de sombreamento. O *Variance Shadow Map* [3] gera sombras suaves a partir do mapa de profundidade da cena em relação à fonte de luz, sem exigir tanto processamento como algoritmos baseados em *Percentage-Closer Filtering*. Sua desvantagem é a presença de “vazamentos” de luz em áreas onde existem sombras de mais de um objeto no mesmo local, porém isto não é um problema para terrenos, já que as sombras do relevo não costumam sobrepor-se. Outra vantagem de sua aplicação em terrenos é que o Sol pode ser considerado uma fonte de luz direcional. Assim, não é necessário calcular a distância euclidiana de cada fragmento até a fonte de luz para obtenção do mapa de profundidade, já que este pode ser obtido diretamente da coordenada *z* dos fragmentos vistos a partir

da posição da luz, que por sua vez é fornecida pelo programa de vértices e interpolada linearmente por fragmento.

2.4. Céu e atmosfera

Para que a cena fique mais realista, precisamos considerar os efeitos causados pela presença da atmosfera. O céu é representado geometricamente por um cubo simples, envolvendo todo o terreno. A aparência do céu é gerada de forma procedural pelo programa de fragmentos aplicado ao cubo. A cor calculada para cada ponto é dada pela posição do Sol e pelo espalhamento de luz na atmosfera.

O desenho do Sol também é gerado proceduralmente, pela Eq. 1, onde \vec{L} é a direção da fonte de luz (Sol) e \vec{V} é a direção da câmera até o fragmento, normalizados, com um certo fator de potência k_i . O produto é feito diversas vezes com diferentes fatores de potência e acumulado na cor final do pixel, para que ocorra uma variação gradual de iluminação do centro do Sol até sua vizinhança. O resultado pode ser visto na Fig. 4a.

$$Sol_{rgb} = \sum_{i=0}^n (\vec{L} \cdot \vec{V})^{k_i} \quad (1)$$

A cor do céu é obtida a partir de três texturas que representam os diversos tons de acordo com a hora do dia, noite e pôr-do-sol (Fig. 4b), simulando o espalhamento de luz na atmosfera. A coordenada de textura *s*, calculada pela Eq. 2, indica a posição longitudinal do Sol dado o ângulo plano entre \vec{L} e \vec{V} , enquanto que a coordenada *t*, descrita pela Eq. 3, onde *h* é um coeficiente de altitude dado pela posição da câmera, indica sua latitude. Em seguida, dependendo da coordenada *y* da direção do Sol, é feita uma interpolação entre as texturas “dia” e “pôr-do-sol” ou “pôr-do-sol” e “noite”.



Figura 4. (a) Representação procedural do Sol. (b) Texturas para dia, pôr-do-sol e noite.

Para tornar o fenômeno de espalhamento da luz mais plausível, o efeito de névoa seca (*haze effect*) é simulado no programa de fragmentos. Uma camada de neblina próxima à superfície contribui para a cor final do fragmento com a mesma cor que seria aplicada ao céu naquele ponto.

Sua intensidade depende da altitude e da distância da câmera até o ponto, calculada pelo programa de vértices e interpolada na rasterização, para melhor desempenho.

$$s = \frac{(\vec{L}_{xy} \cdot \vec{V}_{xy})}{2} + 0.5 \quad (2)$$

$$t = \max(0, V_y)^h \quad (3)$$

2.5. Superfície molhada

Um efeito interessante é o de superfície molhada. Ele pode ser usado quando quisermos simular chuva ou áreas banhadas pelo oceano. Sua implementação é simples: as áreas molhadas recebem um aumento no fator de reflexão espelhada e uma redução no fator de iluminação difusa [4]. A intensidade do efeito pode ser regulada de acordo com o tipo de solo, então as áreas rochosas são as que geralmente sofrem maior variação quando estão molhadas (Fig. 5).



Figura 5. Superfície (a) molhada, (b) seca.

3. Resultados e conclusão

O uso de programação em placas gráficas possibilitou a implementação de técnicas simples para obtenção de uma aparência mais natural e realista. O mapeamento procedural de camadas de textura permite a visualização de terrenos com infinitas variações de relevo. A técnica *Variance Shadow Map* mostrou-se muito adequada para geração de sombras em terrenos (Fig. 6). A nossa solução para simulação atmosférica, com a cor do céu variando em função da hora do dia e com efeito de névoa seca sobre o terreno, é de fácil implementação e melhorou o realismo da cena (Fig. 7).

O programa obteve bom desempenho, com uma média de 270 quadros por segundo¹ para resolução de 1280x1024.

Em trabalhos futuros, a adição de nuvens e efeitos climáticos poderia tornar a cena ainda mais natural [1]. O uso de

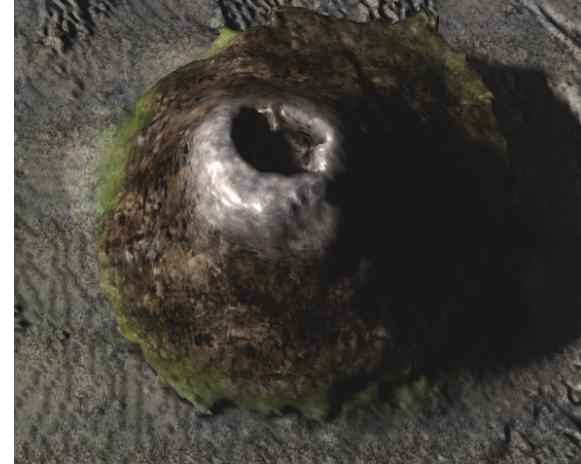


Figura 6. Terreno com sombras suaves.



Figura 7. Dia, pôr-do-sol e noite.

iluminação *High Dynamic Range* (HDR) [2] permitiria efeitos de iluminação mais realistas. Finalmente, para maior desempenho em terrenos muito grandes, técnicas de multirresolução de geometria poderiam ser implementadas.

Referências

- [1] Crytek. “Real-Time Atmospheric Effects in Games, Revisited”. Disponível em <http://developer.nvidia.com/>.
- [2] P. E. Debevec. “Rendering with Natural Light”. *SIGGRAPH*, 1998.
- [3] W. Donnelly and A. Lauritzen. “Variance Shadow Maps”. Technical report, Computer Graphics Lab, School of Computer Science, University of Waterloo, 2006.
- [4] NVIDIA Corporation. “Team Secrets: Cascades”. Disponível em <http://developer.nvidia.com/>.
- [5] NVIDIA Corporation. “Texture Array Terrain”. Disponível em <http://developer.nvidia.com/>.
- [6] K. Perlin. “Improving Noise”. Technical report, Media Research Laboratory, Dept. of Computer Science, New York University, 2002.
- [7] B.-T. Phong. “Illumination for computer generated images”. *ACM Computing Surveys*, 18(6), June 1975.

¹ Testado num Intel Core 2 Duo E6600 com NVIDIA GeForce 8800 GTS 320MB.

Classificação Automática de Imagens do Acervo de Obras de Cândido Portinari

Daniel Pacheco de Queiroz, Antonio da Luz Júnior e Arnaldo de Albuquerque Araújo
Universidade Federal de Minas Gerais - UFMG
Av. Antônio Carlos, 6627 - Pampulha - Belo Horizonte - MG
{pacheco, daluz, arnaldo}@dcc.ufmg.br

Resumo

Métodos eficientes de busca por imagens são cada vez mais necessários e importantes. Nos últimos anos, fatores como a Internet, a disseminação de câmeras digitais, o aumento da capacidade de armazenamento e a diminuição do preço deste armazenamento, fizeram com que a quantidade de informação visual disponível aumentasse consideravelmente. Com isto, surge a necessidade do desenvolvimento de métodos eficientes para recuperação de imagens em grandes bases de dados. Para alcançar este objetivo é necessário descrever o conteúdo das imagens de forma eficaz e de modo que gere dados que possam ser utilizados em uma classificação automática de imagens. Este objetivo é alcançado por meio de atributos relacionados com as características visuais das imagens. Este trabalho visa identificar alguns destes atributos e verificar a eficácia dos atributos selecionados utilizando um classificador automático. Como base foi utilizado o acervo de obras do artista Cândido Portinari, cujas imagens digitais se encontram disponíveis para todos e já possui uma classificação manual, com a qual o método proposto pode ser confrontado.

1. Introdução

O desenvolvimento de sistemas capazes de realizar a tarefa de Recuperação de Imagens Baseado em Conteúdo (Content Based Image Retrieval - CBIR) é a solução ideal para o problema de busca por imagens de interesse em grandes bases de dados [4]. Com este tipo de abordagem é possível recuperar imagens utilizando uma descrição do conteúdo desejado, utilizando para isso um esboço (sketching) ou buscando imagens semelhantes às outras (Query By Example - QBE). O processo de CBIR pode ser dividido em duas etapas principais: a) construção da base de dados; b) recuperação de imagens. Idealmente essas etapas ocorrem em momentos distintos. Na etapa de construção da base de dados, todo conjunto de imagens disponível é processado de

forma a extrair informações, denominadas nesta área como atributos ou características visuais, capazes de identificar o conteúdo de cada imagem. Na segunda etapa, que consiste no processo de recuperação de imagens propriamente dito, é recebido como entrada um conjunto de atributos que descrevem o tipo de conteúdo que se espera encontrar nas imagens. Utilizando-se de técnicas de classificação e medidas de similaridade entre objetos, os dados recebidos são comparados com as imagens previamente obtidas no processamento de um conjunto inicial. Por fim, as imagens são organizadas de acordo com sua similaridade com os atributos de entrada.

A base de imagens das obras de Cândido Portinari, que foi utilizada durante a elaboração deste trabalho, é composta por um conjunto de mais de 5.000 imagens [7]. Estas imagens foram previamente classificadas, em um esforço manual, de acordo com as técnicas e materiais utilizados em suas criações. Como durante a etapa de digitalização das imagens, não foram observados os critérios de qualidade adequados, este trabalho visa auxiliar na classificação das imagens caso seja realizada uma nova etapa de digitalização para corrigir as falhas apresentadas. Para isto é necessário desenvolver métodos capazes de realizar um processo automático de classificação destas novas imagens. Para tal, foi utilizado o conhecimento da categorização das imagens atuais no intuito de construir um classificador eficaz.

Outro objetivo do trabalho é constatar que com o uso de um algoritmo de seleção de atributos, podemos construir classificadores mais eficientes e eficazes, mostrando que nem sempre é melhor extrair o maior número de atributos possível.

2. Referencial Teórico

O primeiro sistema CBIR comercial foi o Query By Image Content (QBIC) [6], desenvolvido pela IBM. Sua análise de atributos é baseada em cor e textura. Alguns sistemas on-line utilizam a engine do QBIC como modelo para suas buscas.

Entre os algoritmos de classificação automática de dados não supervisionada, encontrados na literatura da área, temos o k-Means [11]. Este recebe como entrada uma matriz, as linhas são os itens a serem classificados e as colunas são atributos descritivos, e um número k de categorias. Já entre os algoritmos de classificação supervisionada, temos o Support Vector Machine (SVM) [2] e o k-Nearest Neighbors (k-NN) [1], que calcula a distância de uma imagem de entrada para todas as outras presentes na sua base, ordena as imagens da base por esta distância e define como a classe da imagem de entrada a classe mais freqüente nas k primeiras imagens.

Dentre as abordagens para seleção de atributos determinantes, existem soluções baseadas em algoritmos comerciais como o C4.5 [8] e outras, acadêmicas, como o algoritmo Statistical Association Rule Miner (StARMiner) [10]. O StARMiner foi originalmente utilizado para classificação de imagens médicas. O objetivo deste método é encontrar regras estatísticas envolvendo os atributos que melhor discriminam a imagem em suas categorias. Desta forma, alimentando o algoritmo com as imagens já representadas por seus atributos, ele irá separá-las em categorias e retornará os atributos que foram mais relevantes para a sua classificação.

3. Metodologia

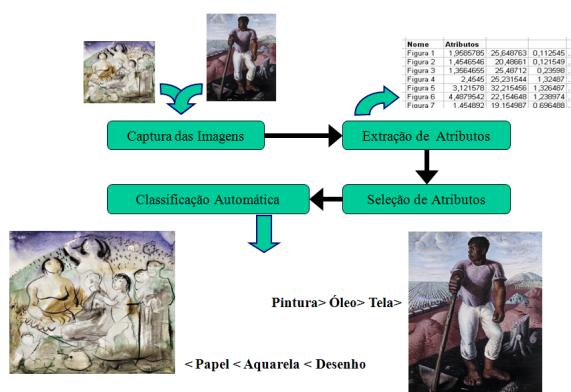


Figura 1. Metodologia

3.1. Captura das Imagens

Inicialmente foi feita a aquisição das imagens que irão compor o universo de pesquisa. Elas estão disponíveis no site do Projeto Portinari [7]. A captura foi realizada acessando-se cada uma das classes de imagens que estão classificadas por suas técnicas. Esta classificação

das obras já existente no site foi mantida, pois será utilizada como referência para quantificar a eficácia da classificação automática gerada neste trabalho.



Figura 2. Menina, pintura a aquarela e grafite sobre cartão, 1941

3.2. Métodos para a Extração dos Atributos

Após a etapa de aquisição de imagens, foi realizada uma pesquisa bibliográfica para identificar quais são os principais métodos utilizados para a extração de atributos visuais, considerando os aspectos de textura e cor.

3.2.1. Baseados em textura. A textura é uma das características mais importantes para a classificação de imagens, e pode ser caracterizada por variações locais nos valores dos pixels que se repetem de maneira regular ao longo da imagem. As texturas podem estar relacionadas a algumas características das imagens como uniformidade, densidade, aspereza, regularidade, intensidade, entre outras.

Como a análise de textura não depende de cor, as imagens foram convertidas para tons de cinza antes da aplicação dos métodos, mantendo-se uma cópia colorida do acervo para a extração dos atributos de cor. Foram selecionados os seguintes métodos baseados em textura:

- Matriz de co-ocorrência
- Transformada de Fourier
- Transformada discreta do cosseno

3.2.2. Baseados em cor

- Histograma normalizado
- Histograma métrico

- Vetor de coerência

3.3. Atributos

A partir da aplicação dos métodos descritos acima, foram extraídos atributos que, segundo a literatura pesquisada, estão associados a características visuais. Os atributos extraídos a partir de cada método estão relacionados abaixo:

3.3.1. Para a matriz de co-ocorrência. Em [5] é descrita uma metodologia de classificação de imagens a partir do uso da abordagem estatística, onde são definidas diversas características advindas do cálculo de matrizes de co-ocorrência, as selecionadas para este trabalho foram:

- Contraste: medida da quantidade de variação local de níveis de cinza em uma imagem.
- Correlação: representa uma idéia de linearidade de dependências de tons de cinza em uma imagem.
- Energia: medida da homogeneidade local dos níveis de cinza em uma imagem.
- Entropia: medida do grau de dispersão de ocorrências de níveis de cinza em uma imagem.
- Variância: medida da dispersão estatística, indicando quanto longe, em geral, os valores se encontram do valor esperado (segundo momento central).
- Diferença entropia
- Diferença variância
- Medida correlação
- Máximo coeficiente de correlação
- Soma da entropia
- Soma da média
- Soma da variância
- Variância inversa

3.3.2. Para a transformada de Fourier e a transformada discreta do cosseno. Baseado em [9], extrairemos os seguintes atributos destas transformadas:

- Primeiro e segundo momentos angulares: o primeiro mede a probabilidade da distribuição dos pixels, enquanto o segundo corresponde à variância.

3.3.3. Para os histogramas. Segundo [3], os histogramas podem ser bem representados por estes atributos:

- Máximo: valor da cor com maior contagem (para cada uma das 3 camadas).
- Mínimo: valor da cor com menor contagem (para cada uma das 3 camadas).
- Média: média das contagens de todas as cores.
- Mediana
- Desvio padrão

3.4. Algoritmo para Seleção dos Atributos Relevantes

Dado o problema da alta-dimensionalidade de atributos a serem analisados, foi adotada uma abordagem para seleção de atributos mais relevantes. Neste trabalho, foi utilizado o algoritmo StARMiner [10] para a mineração de regras de associação estatísticas em imagens. O objetivo do algoritmo StARMiner é encontrar regras estatísticas envolvendo os atributos que melhor discriminam a imagem em suas categorias.

Seja uma base de imagens T , x uma categoria de imagens, $T_x \in T$ o subconjunto de imagens da base da categoria x e A um atributo da imagem, A será relevante se:

$$|\mu_A(T_x) - \mu_A(T - T_x)| \geq \Delta_{\mu min} \quad (1)$$

onde $\mu_{T_x}(Z)$ é a média dos valores do atributo A para o subconjunto de imagens Z e $\Delta_{\mu min}$ é um parâmetro de entrada que indica a diferença mínima permitida entre a média dos valores de A para as imagens da categoria x e a média dos valores de A para as imagens restantes.

3.5. Algoritmo para a Classificação Automática

Para a classificação automática das imagens, foi utilizado o algoritmo de classificação supervisionada k-Nearest Neighbor (k-NN). Este algoritmo realiza a classificação por similaridade. Para isso o vetor de atributos da imagem de entrada é comparado com os vetores de atributos das imagens contidas na base de conhecimento, as quais as categorias são conhecidas. Esta comparação é feita através de uma função de distância (utilizamos a distância euclidiana) que quantifica o quanto perto (ou similar) estão cada par de vetores [1].

A próxima etapa é ordenar as imagens da base de conhecimento, baseando na distância para a imagem de entrada. Desta forma imagens semelhantes, que apresentam distâncias menores, aparecerão no início do vetor ordenado.

A categoria atribuída à imagem de entrada será a mais freqüente nas k primeiras posições deste vetor ordenado. A categoria original da imagem, já conhecida, é mantida para a quantização dos resultados posteriormente.

4. Resultados

Os testes foram realizados utilizando o classificador automático k-NN (com $k=4$), tendo como imagens de sua base metade das figuras de cada classe, escolhidas de uma forma aleatória, mas repetível, para que pudessem ser feitas as comparações entre os testes.

Inicialmente foi feita a classificação sem a seleção dos atributos e posteriormente com a seleção. Com a seleção

a classificação foi testada em três níveis, sendo o terceiro nível o de classificação mais profunda e difícil, pois leva em conta a técnica, o material de pintura e o suporte (ex.: pintura>óleo>tela). O segundo nível considera apenas os dois primeiros níveis, técnica e o material; já o primeiro nível classifica apenas quanto à técnica.

O índice de acerto foi calculado comparando-se a categoria dada pelo classificador automático, com a categoria já atribuída àquela imagem, caso elas sejam iguais configura-se um acerto. Os resultados são os seguintes:

	Sem seleção de atributos	Com seleção de atributos (3 níveis)	Com seleção de atributos (2 níveis)	Com seleção de atributos (1 nível)
Média de atributos por classe	1.342	1.342	1.342	1.342
Média de atributos relevantes por classe	1.342	1.264	1.169	850
Número de classes	220	220	73	8
Número de acertos	370	714	960	2.094
Índice de acerto	14,6361 %	28.5206 %	38.5285%	82.8323%

Tabela 1. Resultados obtidos

5. Conclusões

A melhora observada com a seleção dos atributos determinantes se mostrou bem relevante, alcançando quase 100%. Contudo, analisando os resultados como os de uma máquina automática de classificação, eles estão aquém do que se esperaria.

Este baixo índice de acerto global pode ser justificado por vários fatores, sendo o principal fator a inexistência de um padrão no conteúdo das imagens, imagens com conteúdos completamente diferentes pertencem a uma mesma categoria. Também foram observados problemas com a aquisição das imagens, sendo que algumas foram feitas através de fotos, que englobam fundos e outros objetos que não pertencem às obras. O armazenamento das imagens, em formato JPEG, no qual há perda de qualidade e a geração de macroblocos, também prejudica muito a extração de atributos, principalmente de textura.

6. Agradecimentos

Agradecemos as entidades que apoiaram este projeto: CNPq, CAPES e FAPEMIG.



Figura 3. Exemplo da dissimilaridade de imagens de uma mesma categoria (Gravura > Litografia > Papel)

Referências

- [1] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.
- [2] K.-S. Goh, E. Chang, and K.-T. Cheng. Support vector machine pairwise classifiers with error reduction for image classification. In *MULTIMEDIA '01: Proceedings of the 2001 ACM workshops on Multimedia*, pages 32–37, New York, NY, USA, 2001. ACM.
- [3] W. R. E. Gonzalez, Rafael C. *Digital Imaging processing*. Editora Edgar Blucher Ltda, São Paulo, 2007.
- [4] V. N. Gudivada and V. V. Raghavan. Content-based image retrieval systems. *Computer*, 28(9):18–22, 1995.
- [5] R. M. Haralick, Dinstein, and K. Shanmugam. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3:610–621, November 1973.
- [6] C. W. Niblack, R. Barber, W. Equitz, M. D. Flickner, E. H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. Qbic project: querying images by content, using color, texture, and shape. volume 1908, pages 173–187. SPIE, 1993.
- [7] C. Portinari. Projeto portinari. <http://www.portinari.org.br>.
- [8] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
- [9] T. R. Reed and J. M. H. du Buf. A review of recent texture segmentation and feature extraction techniques. *CVGIP: Image Underst.*, 57(3):359–372, 1993.
- [10] M. X. Ribeiro, J. Marques, A. J. M. Traina, and C. T. Jr. Statistical association rules and relevance feedback: Powerful allies to improve the retrieval of medical images. In *CBMS '06: Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems*, pages 887–892, Washington, DC, USA, 2006. IEEE Computer Society.
- [11] L. Zheng. *Automated feature extraction and content-based retrieval of pathology microscopic images using k-means clustering and code run-length probability distribution*. PhD thesis, Pittsburgh, PA, USA, 2005. Adviser-Paul Munro and Adviser-Michael Becich.

Identificação automática de cromossomos humanos

Guilherme C. Kurtz, Tiago B. Bonini¹, Giovani R. Librelootto (orientador), Luis A. Perles,
Michele R. Sagrillo

UNIFRA – Centro Universitário Franciscano, Rua dos Andradas, 1614, Santa Maria, RS
[{giovani,perles,sagrillo}@unifra.br](mailto:{guitarro17,tiagobonini}@gmail.com)

Abstract

Karyotyping is the chromosome classification and identification. Such task may take some time. With a picture of chromosomes is possible to automatize such process. One of the first steps is an image segmentation where all chromosomes are extracted from the image. To do this, a sequence of blurring, edge detection, erode/dilate is applied to the image to make a mask. The last step creates islands inside some chromosome mask leading to data loss. An algorithm has been developed to delete those islands. A previous step for the classification has been implemented using the position of the centromere determined by detection of the narrow part of the chromosome. Further developments include arm proportion and banding pattern recognition systems.

Resumo

Cariotipagem é a classificação e identificação dos cromossomos. Tal tarefa leva algum tempo. Com uma imagem de cromossomos é possível automatizar tal processo. Um dos primeiros passos é a segmentação da imagem onde todos os cromossomos são extraídos da imagem. Para isso, uma seqüência de suavização, detecção de bordas, erosão e dilatação é aplicada a imagem para fazer uma máscara. O último passo cria ilhas em alguns cromossomos da máscara levando a perda de dados. Um algoritmo foi desenvolvido para deletar essas ilhas. Um passo anterior da classificação foi implementado usando a posição do centrômero determinada pela detecção da parte mais estreita do cromossomo. Desenvolvimentos futuros incluem a proporção dos braços e sistemas de reconhecimento de padrão de bandas.

1. Introdução

O desenvolvimento da tecnologia tem possibilitado a exploração criteriosa de diversos segmentos relacionados à área da ciência que estuda a cadeia

DNA e cromossomos, denominada citogenética. Com isso, têm sido oferecidos no mercado mundial alguns sistemas computadorizados dedicados à análise cromossômica. Estes sistemas apresentam funções tais como a geração da representação dos cromossomos em pares (pareamento) de acordo com padrões internacionais, conhecidos como Cariótipo.

Apesar da demanda crescente, é pequena a oferta de sistemas automáticos que auxiliem o trabalho dos geneticistas na coleta de dados e geração do Cariótipo. Portanto, o objetivo do presente trabalho é desenvolver um sistema computadorizado utilizando técnicas de visão por computador capaz de auxiliar o geneticista tanto na execução da análise da forma do cromossomo humano a fim de realizar o pareamento assistido dos cromossomos como também auxiliar na coleta e análise de dados para realização do estudo do cariótipo [2].

Para criar um sistema que identifique cromossomos automaticamente, optou-se pelo processamento de imagens. Essa área tem como objetivo a melhora do aspecto visual de imagens, além de fornecer métodos para a sua interpretação, possibilitando a submissão para posteriores etapas de processamento. Detalhes sobre este tema são descritos na seção 2.

Para efetuar um comparativo da abordagem utilizada com outras soluções existentes, a seção 3 apresenta trabalhos relacionados. A seção 4 apresenta a metodologia proposta para a identificação automática de cromossomos. Por fim, a conclusão do artigo é efetuada na seção 5.

2. Processamento de imagens

O processamento de uma imagem digital abrange tanto fundamentos teóricos quanto a parte de software e de hardware e é dividido em cinco etapas: a aquisição da imagem, o pré-processamento, a segmentação, o reconhecimento e a interpretação [3]. Dentre estas, as que serão abrangidas neste trabalho são as etapas de pré-processamento e segmentação, principalmente.

¹ Aluno bolsista de PROBIC – UNIFRA.

Após obter uma imagem digital, é necessário fazer o seu pré-processamento e em seguida sua segmentação, de modo a utilizar-se de técnicas para o realce de contraste e brilho e remoção de ruídos. Essas técnicas se dividem em duas categorias: métodos no domínio espacial e métodos no domínio de freqüência [3]. O *domínio espacial* refere-se à manipulação direta dos *píxeis* da imagem. Técnicas no *domínio de freqüência* são baseadas nas modificações através das transformadas de Fourier. O trabalho desenvolvido utilizou técnicas no domínio espacial.

Alem disso, é necessário fazer a separação das regiões de interesse da imagem (segmentação). A saída do estagio de segmentação constitui-se de dados em forma *píxeis*, que corresponde tanto à fronteira de uma região como a todos os pontos dentro da mesma.

3. Trabalhos Relacionados

A ferramenta KEI [4] utiliza uma técnica de detecção de bordas para elaborar a segmentação da imagem. É possível que o operador altere o resultado da segmentação caso a imagem não traga um resultado final satisfatório. Após a segmentação e cariotipagem, o sistema apresenta imagens dos cromossomos classificados, permitindo ao usuário efetuar trocas entre eles caso o resultado não esteja totalmente correto.

Além disso, podemos citar o que foi desenvolvido por [7], no qual se utilizou a técnica de aprendizado com redes neurais para diminuir a taxa de erros na classificação de cromossomos. Neste trabalho, testaram-se diversas arquiteturas de redes neurais, visando comparar o resultado obtido com outras abordagens, como Algoritmos de Busca em Grafos, Redes de Markov e os Algoritmos de Transportes.

O Kario foi criado para facilitar o trabalho dos geneticistas no reconhecimento do cariótipo [8]. O Kario utiliza um banco de dados MS Access para o armazenamento dos dados textuais, sobre os quais ele permite que se façam consultas. Entretanto, tal como o KEI, o reconhecimento dos cariótipos não é 100% fiel.

3. Identificação Automática de Cromossomos

A análise e identificação de cromossomos humanos são chamadas de cariotipagem. A espécie humana possui 23 pares de cromossomos, sendo 22 pares autossomos e um par de cromossomos sexuais (XX ou XY). A automação deste processo torna-se útil visto que as imagens obtidas geralmente são de baixa qualidade, dificultando sua classificação.

O ponto de partida neste projeto é uma imagem digitalizada de um conjunto de cromossomos humanos, como demonstrado na figura 1. Esta imagem passará por etapas de pré-processamento e segmentação, podendo ter interferência direta de um operador, caso seja necessário. Após essas etapas, a imagem é submetida a um algoritmo de reconhecimento de padrões [5], que terá como critério de classificação o tamanho e o padrão de bandas dos cromossomos.



Figura 1. Imagem de cromossomos humanos

A partir da imagem original, cria-se uma nova imagem chamada de máscara (seção 4.1). Essa máscara será uma imagem binária obtida através de processos de suavização, limiarização, dilatação, erosão e remoção de ilhas. A partir da imagem pré-processada, a segmentação tratará de separar cada uma das regiões (cada cromossomo) e armazená-las separadamente.

Logo, o reconhecimento de padrões utiliza-se de ideogramas como modelos para a tentativa de reconhecimento dos cromossomos. Duas técnicas foram implementadas: o reconhecimento por tamanho e o reconhecimento por padrão de bandas. Por fim, exibe-se a seqüência dos cromossomos classificados.

3.1. Pré-processamento

Neste trabalho, a etapa de pré-processamento envolve a utilização de técnicas de realce de contraste, suavização, dilatação, erosão. Estas técnicas visam como resultado final uma imagem monocromática, destacando os cromossomos da imagem com a cor branca e as regiões que não interessam com a cor preta.

A suavização de imagens é aplicada com o fim de eliminar os ruídos, reduzindo detalhes que não são desejados. Para isto, é aplicado um filtro espacial de passa baixa, utilizando *vizinhos-de-8* [3]. O resultado da aplicação deste filtro será uma imagem com menos ruídos, o que facilita um realce de contraste mais uniforme sem extrapolar os limites desejados.

Neste caso, dado um intervalo em níveis de cinza, todos os *píxeis* que contenham um valor que pertença a este intervalo terão seu contraste ajustado para o branco total (nível 255) ou preto (nível 0). Entretanto, somente técnicas de suavização e realce de contraste

não são suficientes para se obter uma imagem capaz de destacar os cromossomos de uma imagem.

Para facilitar a visualização e a definição das regiões procuradas, aplica-se uma dilatação sobre a imagem, eliminando lacunas. Por sua vez, a erosão de uma imagem serve para a eliminação de detalhes que são irrelevantes em uma imagem.

Além das transformações de erosão e dilatação, desenvolveu-se uma nova transformação com base nesses dois, que será chamado de remoção de ilhas. Por exemplo, a figura 2 apresenta vários *píxeis* da com nível de cinza igual a 0 (ou *píxeis* pretos) na imagem. Se este *píxel* estiver em uma ilha de *píxeis* pretos cercados de *píxeis* de nível de cinza igual a 255 (ou *píxeis* brancos), todos estes *píxeis* deverão ter seu valor de nível de cinza alterado de 0 para 255, pois estes constituem de um buraco na máscara do cromossomo.



Figura 2. Exemplo da imagem antes e depois da aplicação da remoção de ilhas

Com estes filtros e transformações, a imagem original é mantida intacta e é criada a uma nova imagem contendo uma máscara sobre a original, como se percebe na figura 3.



Figura 3. Máscara

A partir de então, a etapa de segmentação irá trabalhar na separação de cada um desses cromossomos com a finalidade de iniciar o processo de identificação dos mesmos.

3.2. Segmentação

A partir de uma imagem pré-processada, o objetivo da segmentação é obter um conjunto de segmentos separados com informações em relação à imagem de origem. A segmentação utilizada foi por regiões, as quais são os cromossomos separados.

Após o pré-processamento, faz-se uma varredura pela máscara na busca por estes *píxel* com nível de cinza igual a 255 (branco total): a partir de um *píxel* encontrado com nível de cinza igual a 255, todos os *píxeis vizinhos-de-4* são selecionados e armazenados

até que se encontre um novo *píxel* com valor 0. A partir destes *píxeis* armazenados, o processo se repete até que todos os *píxeis* desta região de interesse tenham sido armazenados. Após isso, o processo se repete na tentativa de encontrar uma nova região de interesse.

Logo, todas as regiões que contenham pontos interligados de níveis de cinza com valor de 255 são retiradas da imagem original (tendo como referência a máscara) e armazenadas separadas em um vetor que contem os *píxeis* de cada região e sua posição na respectiva imagem. A Figura 3 apresenta a máscara da imagem original (Figura 1) obtida após o seu pré-processamento e pronta para ser usada como referência na segmentação. Ao utilizar uma imagem binária como máscara, as regiões de interesse que devem ser segmentadas ficam limitadas aos *píxeis* de nível de cinza igual a 255, e as regiões de não interesse com nível de cinza igual a 0.

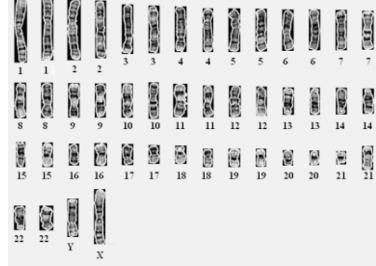


Figura 4. Resultado final após a identificação dos cromossomos da Figura 1

Por fim, tem-se armazenado todas as regiões de interesse da imagem separadas, possibilitando o reconhecimento de padrões. Porém, nessa fase surgem diversos problemas, tais como imagens com cromossomos tortos, inclinados, ou até mesmo sobrepostos. Os problemas de inclinação são ajustados após os cromossomos estarem segmentados, através da rotação dos mesmos. Com um pré-processamento bem elaborado, alguns destes problemas são resolvidos sem a necessidade de ajustes futuros, facilitando a identificação dos cromossomos.

3.3. Reconhecimento de padrões e identificação dos cromossomos

Dentre as características utilizadas na cariotipagem, o método aqui proposto utiliza inicialmente o tamanho do cromossomo e o seu padrão de bandas[1]. Para ambas as características avaliadas, foram utilizadas imagens padrões na forma de ideogramas. Os ideogramas contêm o padrão de bandas de todos os cromossomos e o seu tamanho proporcional em relação aos outros. Isso possibilitou a utilização de um único conjunto de objetos que serve para os dois casos de avaliação, facilitando uma futura mixagem dos

métodos. Na figura 5 percebe-se uma imagem contendo os modelos utilizados nesta abordagem.

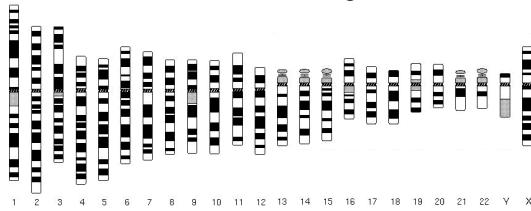


Figura 5. Ideograma

Na cariotipagem baseada no comprimento, é realizada uma medida longitudinal do cromossomo. Cada cromossomo na célula tem um comprimento único [1], o que possibilita a classificação dos mesmos utilizando métodos baseados no comprimento. O processo é comparar cada ideograma com todos os cromossomos de acordo com a proporção de seu tamanho com a proporção do tamanho dos demais cromossomos na célula. Então, é realizada uma busca visando a menor diferença possível entre os ideogramas e o cariotípico.

O padrão de bandas possibilita combinações diversas no processo de classificação da amostra [1]. Utilizou-se o padrão de bandas através de métodos no conjunto espacial. A busca se dá na seguinte forma: cada ideograma é comparado com todos os cromossomos. Nestes, é feita uma mudança de escala de acordo com o tamanho longitudinal do ideograma. Então, calcula-se o somatório da diferença dos níveis de cinza da imagem do ideograma (i) com a imagem o cromossomo (p), sendo o resultado final o índice de semelhança entre ambos:

$$\sum_{j=0}^h |p(w/2, j) - i(w/2, j)|$$

O cálculo é feito a partir da coluna do meio da imagem (sendo w a largura da imagem, $w/2$ são os píxeis do meio). A menor diferença dará a classificação do cromossomo como sendo o mesmo do ideograma que foi comparado.

5. Conclusão

Atualmente, existem diversas técnicas de processamento de imagens para o reconhecimento de padrões cromossômicos. Porém, nenhuma delas retorna um resultado considerado ótimo [1]. Se por um lado, a utilização de técnicas de aprendizado traz uma resposta melhor a este problema, por outro, a utilização das mesmas pode se tornar um processo demorado e com alto custo, como o uso de técnicas de aprendizado com redes neurais para a busca da classificação dos mesmos [7]. Entretanto, por vezes é necessária a utilização de técnicas menos exatas, mas com um custo inferior.

Desta forma, o presente artigo abordou a construção de uma ferramenta de identificação automática de cariotípicos. A metodologia envolveu praticamente todas as etapas do processamento de imagens, de forma a permitir o reconhecimento de cromossomos humanos, inclusive, em imagens de baixa qualidade.

A ferramenta atual possui mecanismos para manipular a imagem antes das etapas de segmentação, tais como separação de regiões que não deveriam estar unidas, recorte, dilatação, erosão, entre outras.

Das técnicas utilizadas para o processo de identificação dos cromossomos, a classificação por tamanho (demonstrada na figura 5) teve um desempenho melhor. Existe a possibilidade de utilizar os pontos positivos de cada uma delas para criar um novo método de reconhecimento.

Para a intervenção do operador, desenvolveu-se uma interface que permite efetuar os ajustes na máscara da imagem, caso necessário. Esta intervenção auxilia nas etapas de segmentação e identificação. Entretanto, o objetivo é reduzir o trabalho do operador e tornar o processo o mais automático possível.

Os próximos passos envolvem o tratamento de cromossomos tortos e sobrepostos, bem como uma bateria de testes sobre o sistema envolvendo imagens das mais diversas qualidades, contendo uma variada disposição dos cromossomos nas mesmas.

6. Referências

- [1] Biyani, P., Wu, X., and Sinha, A.(2005) "Joint Classification and Pairing of Human Chromosomes". IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol. 2, No. 2, April-June.
- [2] Carothers, A., and Piper, J (1994). "Computer-aided classification of human chromosomes: a review". Journal Statistics and Computing. Springer. Vol. 4, No. 3.
- [3] Gonzales, R. C., Woods, R. E., Junior, C. (2000) "Processamento de imagens digitais". Blucher.
- [4] Qualiterm Eletrônica (2003). "Key: Editor de Cariótipos Profissional". <http://www.qualiterm.com.br/folhetos/key.pdf>
- [5] Schalkoff, R. (1992). Patter Recognition – Statistical, Structural and Neural Approachs, John Willey & Sons.
- [7] Todesco, J. L. (1995). Reconhecimento de Padrões Usando Rede Neural Artificial com Função de Base Radial: Uma Aplicação na Classificação de Cromossomos Humanos. Tese de Doutorado, UFSC.
- [8] ZETA Imaging (2008) "Kario Ver. 8.8". <http://imageanalysis.altervista.org>

Visão Computacional aplicada à Localização e Mapeamento Simultâneos de Robôs Subaquáticos

Paulo Drews Jr , Silvia Botelho

Fundação Universidade Federal do Rio Grande - Departamento de Física - NAUTEC
Av. Italia Km 8 - Rio Grande - RS - Brasil
paulo,silviacb@ee.furg.br

Resumo

O uso de veículos subaquáticos autônomos para inspeção visual é um campo promissor da robótica. Devido à dificuldade de localizar o robô e mapear o ambiente simultaneamente (SLAM), este trabalho propõe o uso de visão computacional e de mapas topológicos. Utilizando uma câmera de vídeo como fonte sensorial, essa abordagem é composta por dois estágios principais: i) aplicação do SIFT para extração de características em seqüências de imagens e ii) uso de mapas auto-organizáveis. O sistema desenvolvido foi validado em situações reais e simuladas, usando robôs reais em testes online. A precisão e a robustez obtidas em condições subaquáticas desfavoráveis, como variação de iluminação e ruído, conduzem a uma original e eficiente técnica de SLAM.

1. Introdução

Sendo o ambiente marinho rico em recursos biológicos e minerais, assim sua exploração é relevante e atrativa. Avanços tecnológicos vêm conduzindo ao desenvolvimento de veículos autônomos subaquáticos (*autonomous underwater vehicles* - AUVs) para realização das mais diferentes tarefas [9]. Devido às restrições associadas ao ambiente subaquático, bem como à longa duração das missões, busca-se diminuir a interferência humana na realização das atividades do veículo. Diversos tipos de sensores vêm sendo utilizados de forma a permitir uma percepção eficiente do meio, todos eles de custo elevado, tais como *Doppler Velocity Log* (DVL), *Acoustic Transponder Networks* (ATN) e *Inertial Navigation Systems* (INS) [3]. Entretanto, o conjunto de incertezas relacionadas ao meio ainda impedem a existência de uma solução de alto desempenho a custo exequível.

Por outro lado, AUVs quando utilizados para inspeção visual são equipados com uma câmera presa ao robô [3]. Essa câmera realiza a captura de imagens do fundo do mar, por meio das quais podem ser determinados marcos naturais (características naturalmente presentes no meio). Os

marcos são representados por descritores que compõem os mapas do ambiente navegado, viabilizando a localização e mapeamento simultâneos do robô (*simultaneous localization and mapping* - SLAM) ao longo da sua navegação. Assim, frente ao desafio, não trivial, associado ao problema de SLAM subaquático, o uso de informação visual surge como uma solução criativa, robusta, de baixo custo e de bom desempenho.

2. Trabalhos Relacionados

A proposta aqui apresentada é inovadora, embora existam diversos trabalhos na área de SLAM visual subaquático. Muitos dos trabalhos existentes são baseados em mosaicos. Um dos pioneiros em mosaicos subaquáticos, [8] propõe a busca de contornos por meio do laplaciano da gaussiana (LoG). A correlação por meio desse operador permite maior tolerância à iluminação não-uniforme. Embora o sistema desenvolvido operasse em tempo real, o movimento do robô era restrito a uma trajetória em colunas previamente conhecidas ("zigzag"). Outro trabalho que permitiu localização e mapeamento utilizando mosaicos, foi desenvolvido por [10]. Esse detectava características por meio do método KLT.

Embora tenham sido usados os mais diversos métodos para detecção de características em ambiente subaquático, não foram encontrados trabalhos usando o SIFT com essa finalidade. Uma abordagem com SIFT para SLAM visual foi realizada por [11]. Estes usaram o SIFT em um sistema visual estéreo para determinar marcos visuais, juntamente com a informação de odometria. Os testes foram feitos em ambiente terrestre estruturado com mapas conhecidos *a priori*. [2] apresentam a abordagem mais semelhante à proposta por este trabalho. Tais autores realizam SLAM visual com mapas topológicos baseados em aparência. Para isso, o método SIFT é usado em imagens omnidirecionais. Esta abordagem foi validada com um robô móvel em ambiente terrestre estruturado.

3. Sistemas Visuais em Ambiente Subaquático

Mesmo que se identifique diversas situações de aplicação do sistema aqui proposto, algumas restrições precisam ser respeitadas para a sua utilização, são elas: *i*) a navegação do veículo deve ser, aproximadamente, restrita a um plano, por exemplo, paralela ao fundo do mar, ou paralela a um casco de navio, com a câmera de vídeo orientada para tal plano; *ii*) a turbidez da água deve permitir uma visibilidade mínima; *iii*) assume-se que o terreno subaquático é razoavelmente plano, ou seja, tenha desniveis desprezíveis com relação à distância da câmera ao solo; *iv*) a condição de iluminação deve ser adequada; *vi*) a imagem precisa ter características distintivas mínimas; *vii*) o ambiente a ser explorado não pode ser excessivamente dinâmico, de forma a impossibilitar a inferência de movimento entre imagens.

4. Um Sistema para SLAM Visual

O sistema proposto neste trabalho é composto por diversos módulos, como mostrado na figura 1. Cada módulo será detalhado nas seções a seguir.



Figura 1. Overview do sistema proposto.

4.1. Pré-Processamento

As distorções causadas pelas lentes da câmera podem ser representadas por uma aproximação radial e tangencial. Como a componente radial causa a maior distorção, a maioria dos trabalhos desenvolvidos somente corrige esta componente [4] [3]. Em ambientes subaquáticos surge mais uma distorção, causada pela difração tanto na água como dentro da própria câmera [12]. Para um melhor desempenho, esta abordagem é implementada em uma LUT (*Lookup Table*).

4.2. SIFT

O SIFT é um método robusto para extrair e descrever características de uma imagem [7]. O algoritmo é composto

por 4 etapas. A **detecção de extremos no espaço de escala** na qual é feita a busca de características em todas as escalas e localizações de imagens com diferença de filtros gaussianos (DoG), estas invariante a escala. A **localização de pontos-chaves** onde é determinada a localização exata e escala de cada *keypoint*, nessa etapa são definidos os melhores pontos para o mapeamento através do gradiente. Também é feita a **definição da orientação** para cada *keypoint* por meio dos gradientes locais ao ponto, obtendo-se invariância a rotação. Por fim, são determinado os **descritores dos pontos-chaves** onde o gradiente local de cada ponto-chave é medido, utilizando-se a vizinhança do ponto. Estas medidas são transformadas em uma representação que permite tolerância a níveis significativos de distorção e mudança de iluminação.

4.3. Matching e Estimação da Matriz de Homografia

O melhor candidato para correlacionar cada ponto-chave é encontrado pela identificação de seu vizinho mais próximo. Entretanto, muitos pontos de uma imagem não têm correspondentes na imagem anterior. Para eliminar esses falsos *matches*, utiliza-se um método que compara a menor distância com a segunda menor distância [7]. Além disso, o sistema usa a estimativa da matriz fundamental para remover *outliers*, ou falsos verdadeiros. Essa matriz é estimada pelos métodos robustos RANSAC e LMedS, separadamente. Então, são usados os resultados obtidos por ambos os métodos, para validar cada correlação.

A correlação entre pontos de duas imagens distintas permite a determinação do movimento relativo entre elas. Os *n* pares de pontos são usados para determinar a matriz de homografia, H , que permite estimar o movimento entre estas imagens [5].

4.4. Mapas Topológicos

Neste trabalho, é proposto o uso de mapas topológicos, ou seja, mapas que baseiam-se na noção de um espaço não quantitativo. Eles são construídos usando os descritores dos pontos extraídos pelo SIFT e a pose adquirida da homografia. Tais informações são usadas em mapas auto-organizáveis, baseados nas redes de Kohonen [6]. A estrutura do mapa consiste de nodos, ou neurônios, tais conectados por arestas, que representam a idéia de vizinhança. Associado com cada neurônio existe um vetor de pesos, que tem a mesma dimensão do descritor do SIFT, além da posição e orientação.

Durante a navegação, os descritores são apresentados ao mapa. Quando um novo caso de treinamento é apresentado, o mapa determina se um neurônio deve ser criados ou atu-

alizado, eles ainda podem ser excluídos ou confirmados. O mapa topológico, depois de suficientemente treinado, permite a navegação de duas formas: por posições objetivo ou por marcos visuais. A partir da posição atual do robô, um algoritmo de busca em grafos Dijkstra ou algoritmo A^* , é usado para buscar a posição alvo, dependendo do desempenho desejado e do tamanho do mapa.

5. Implementação, Testes e Resultados

O presente trabalho foi totalmente implementado, sendo validado a partir de um conjunto de testes envolvendo missões simuladas e reais. O sistema foi acoplado ao veículo ROVFURG-II apresentado na figura 2. Esse robô é equipado com uma *Tritech Typhoon*, um sonar *Tritech MiniKing* e um conjunto de sensores (altímetro e acelerômetros). O sistema foi testado em um computador Intel Core 2 Quad Q6600 com 2Gb de RAM DDR2-667. A câmera usada tem padrão NTSC, porém utilizou-se frames de 320x240 com uma taxa de até 29,97 fps.

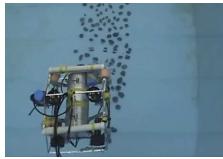


Figura 2. ROVFURG-II durante testes.

Visando verificar a robustez da proposta, frente a diferentes condições do meio, foram aplicados diferentes filtros às imagens [1], de forma a enaltecer determinadas características subaquáticas, como turbidez, neve marinha, iluminação não-linear e outras. A tabela 1 apresenta os diferentes filtros aplicados.

Característica / Filtro	1	2	3	4	5
Distância da Fonte de Luz (m)	0.2	0.22	0.25	0.25	0.3
Atenuação (%)	0.05	0.05	0.06	0.05	0.05
Ruído Gaussiano (σ)	2	2	2	4	4
Mínimo Nível de Cinza	20	30	20	20	20
N.º de Flocos de Neve Marinha	30	30	30	30	30

Tabela 1. Características subaquáticas para cada distorção usada nos testes.

5.1. Localização em diferentes condições subaquáticas

O sistema visual foi testado usando as cinco diferentes condições de ambiente subaquático, correspondendo a cada

(**filtro i**) apresentado na tabela 1. A figura 3, na parte superior, enumera os pontos detectados e correlacionados obtidos em uma missão. Embora o número de pontos e de correlações tenham diminuído com a perda de qualidade devido aos filtros aplicados à imagem, ainda assim foi possível localizar o robô, como mostra a figura 3, na parte inferior. Nesta figura, o movimento de referência é apresentado em azul, executado por um braço robótico composto por um atuador do tipo *harmonic drive* com encoder acoplado, fornecendo uma informação angular a cada 0,000651 segundos, com uma câmera acoplada. Nota-se que a abordagem proposta é robusta à mudanças de condições do ambiente subaquático.

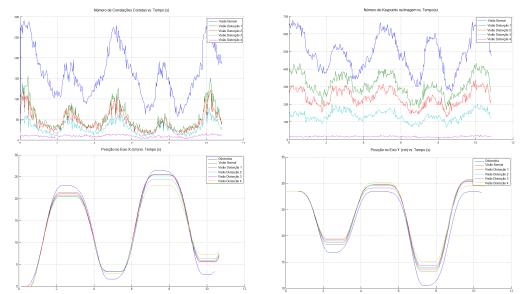


Figura 3. Número de *keypoints* e correlações verdadeiras, além das posições nos eixos euclidianos, durante movimento de um braço robótico, frente a diferentes filtros.

5.2. Localização Robótica *Online*

Foram realizados testes para avaliar o desempenho do SIFT, considerando como comparação um algoritmo clássico na localização robótica em ambiente subaquático, o KLT [10]. Em tais testes, utilizou-se a metodologia completa, apenas mudando o algoritmo localização visual. A figura 4 mostra o resultado de desempenho dos algoritmos. A posição final está representada por um 'X' verde e as medidas em centímetros. SIFT obteve uma taxa média de 4,4 fps (frames por segundos), sem distorção, e 10,5 fps com o **filtro 5**. O KLT apresentou maiores taxas de frames processados, 13,2 fps e 13,08 fps, respectivamente. Vale notar que o SIFT apresentou um melhor desempenho quando as imagens tinham qualidade inferior, isto se deve ao fato do custo computacional estar relacionado, principalmente, com a determinação dos descritores locais dos pontos, que estão em menor número nesse caso. No entanto, como a dinâmica associada com o movimento é lenta, ambos os métodos podem ser aplicados ao SLAM *online*.

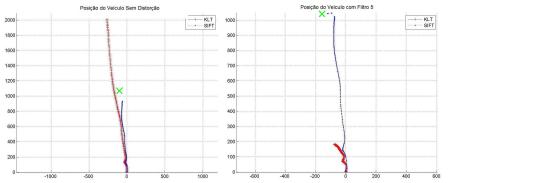


Figura 4. Localização *online* do robô, com e sem distorção.

5.3. Mapas Topológicos Auto-organizáveis

Testes para validar o sistema de mapeamento proposto foram realizados. A figura 5 mostra o mapa final, usando imagens adquiridas durante a navegação do veículo. Tal mapa pode ser usado para localizar o veículo, e auxiliar a navegação por meio de alvos visuais além de poder ser utilizado para corrigir a pose do robô, utilizando informações previamente apresentadas ao mapa, validando o SLAM e questões online associadas à tarefa de inspeção do AUV.

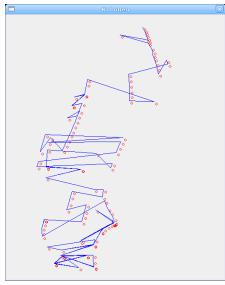


Figura 5. Mapa topológico gerado pelo movimento do ROVFURGII.

6. Conclusão

O trabalho apresentou uma nova abordagem para SLAM subaquáticos, utilizando apenas a informação visual *online*, facilmente extensível a outros ambientes, como aéreo e subterrâneo. Tal sistema pode ser utilizado em tarefas de inspeção autônomas, ou no auxílio ao controle do robô. Diversos testes foram realizados com diferentes condições subaquáticas. A efetividade da proposta foi validada em um conjunto de cenários reais. Os resultados mostraram a vantagem de utilizar SIFT, em relação a outros métodos tradicionais como o KLT, devido à sua invariância à escala e rotação, além desse ser tolerante a variações de iluminação e perspectiva. O mapeamento topológico, apesar de pouco difundido, também mostrou potencial para a aplicação, sendo

robusto a imprecisões sensoriais além de apresentar baixo custo computacional e possibilitar a correção da pose do robô.

Como trabalhos futuros, é proposta uma análise detalhada do sistema de mapeamento topológico, executando um conjunto de testes com diferentes cenários e parâmetros. Também é proposto o uso da informação de escala fornecida pelo SIFT em conjunto com a informação do altímetro do veículo, permitindo a estimativa do movimento de profundidade do veículo. A utilização de visão estereoscópica também é uma possibilidade, podendo conduzir a melhores resultados e maior aplicabilidade do método.

7. Agradecimentos

O trabalho apresentado está relacionado ao Projeto TEC-SUB financiado pelo CNPq, e foi inteiramente desenvolvido pelo aluno ao longo da duração de sua bolsa. Agradece-se o auxílio experimental dos pesquisadores dos laboratórios NAUTEC e NuMA da FURG.

Referências

- [1] M. Arredondo and K. Lebart. A methodology for the systematic assessment of underwater video processing algorithms. In *IEEE/OES Oceans*, pages 362–367, 2005.
- [2] O. Booij, B. Terwijn, Z. Zivkovic, and B. Kroese. Navigation using an appearance based topological map. In *IEEE ICRA*, pages 3927–3932, April 2007.
- [3] R. Garcia. *A Proposal to Estimate the Motion of an Underwater Vehicle Through Visual Mosaicking*. PhD thesis, Universitat de Girona, 2001.
- [4] N. Gracias, S. Van der Zwaan, A. Bernardino, and J. Santos-Vitor. Results on underwater mosaic-based navigation. In *MTS/IEEE Oceans*, volume 3, pages 1588–1594, 10 2002.
- [5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
- [6] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
- [7] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [8] R. Marks, S. Rock, and M. Lee. Real-time video mosaicking of the ocean floor. *IEEE Journal of Oceanic Engineering*, 20(3):229–241, July 1995.
- [9] J. R. McFarlane. Underwater technology 2000 rovs and auvs: tools for exploring, exploiting and defending the ocean frontier. *Underwater Technology*, pages 465–471, 2000.
- [10] K. Plakas and E. Trucco. Developing a real-time, robust, video tracker. In *MTS/IEEE Oceans*, pages 1345–1352, 2000.
- [11] S. Se, D. Lowe, and J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364–375, June 2005.
- [12] X. Xu and S. Negahdaripour. Vision-based motion sensing for underwater navigation and mosaicing of ocean floor images. In *MTS/IEEE Oceans*, pages 1412–1417, 1997.

Extração Automática de Características em Imagens de Impressão Digital

Eduardo da Silva Ribeiro, Erikson Freitas Moraes (*orientador*) e Gilberto Luis Valente da Costa

Faculdades Integradas de Caratinga

Caratinga – Minas Gerais - Brasil

{eduardosss,eriksonfm,betogil}@gmail.com

Resumo

Esse artigo trata-se de um estudo onde foi investigado uma metodologia para a extração de características em imagens de impressão digital. O principal objetivo foi selecionar métodos que fossem eficientes para o processamento de imagens de impressões digitais originados por equipamentos de diversas tecnologias. Na etapa de segmentação foi escolhido um método que utiliza o cálculo da variância. O filtro de Gabor foi selecionado para a melhoria da qualidade da impressão digital e foi usado um método para a obtenção do mapa de direções para este filtro. Para a extração de características, conhecidas como minúcias, utilizou-se o filtro de afinamento de Zhang e Suen e o algoritmo Cross-number. Para os testes foi escolhido o banco de dados de imagens da competição FVC e imagens obtidas a partir de um leitor Digital Persona.

1. Introdução

Com a popularização dos coletores de impressão digital, esta passou a ser utilizada em aplicações comerciais e financeiras. Seu uso anterior era mais restrito a investigações criminais e identificação civil. Segundo [7], a impressão digital é a que melhor atende os requisitos biométricos, por isso é a mais usada em sistemas de identificação por biometria. Entretanto, no mercado existem muitos tipos de coletores de impressão digital, que empregam tecnologias distintas e geram imagens com características diferentes. Cada fabricante disponibiliza um *software* que faz o reconhecimento da impressão digital do seu leitor, mas estes *softwares* são geralmente de código proprietário e utilizados em aplicações restritas.

A idéia principal deste trabalho é utilizar um método de processamento de imagem que seja eficaz em processar imagens de impressões digitais de

leitores diferentes, utilizando o mesmo *software*. O trabalho desenvolvido tem o objetivo de iniciar um projeto maior que consiste em projetar um sistema de autenticação de usuários feito pela impressão digital para uso na plataforma *Web*.

2. Impressões Digitais

As impressões digitais são elevações da pele existentes na ponta dos dedos. O desenho formado por essas elevações são diferentes para cada indivíduo, ou seja dois indivíduos não possuem exatamente igual a mesma impressão digital[1]. Historiadores descobriram que os egípcios antigos já utilizavam a impressão digital como forma de identificação. Entretanto somente no final do século XIX foi feito um estudo científico da impressão digital, e esta passou a ser utilizada como meio para comprovar a identidade de uma pessoa[1,7].

Existem algumas formas de fazer a identificação de uma pessoa através da impressão digital, uma delas é através das relações que existem entre as minúcias das impressões digitais. As minúcias são acidentes que ocorrem com as cristas das digitais. Existem vários tipos de minúcias mas as encontradas com maior freqüência são as terminações e bifurcações [5]. Esses dois tipos de minúcias foram utilizadas no trabalho.

3. Metodologia

A análise bibliográfica realizada foi importante para a seleção de métodos de processamento de imagem utilizados, considerou-se principalmente os trabalhos de [2,8,9,10]. Em seguida, foi proposto uma série de etapas para o sistema de detecção de minúcias. Para segmentar escolheu-se utilizar o cálculo da variância e para a melhoria da qualidade da imagem utilizou-se o filtro de Gabor com um mapa direcional. A etapa seguinte consistiu em aplicar um filtro de

afinamento e o algoritmo para a detecção das minúcias.

3.1. Segmentação

A segmentação de uma imagem tem o objetivo de separar a região de interesse da imagem das demais. Geralmente consiste em discernir um objeto do fundo[6]. No caso de uma imagem de impressão digital a segmentação separa a região onde está a impressão digital do fundo. Este tende a ser constante ou ter uma variação de intensidade de cor pequena. Já a região onde está localizada a impressão digital tem uma variação de cor maior porque as cristas e os vales da possuem um variação de cor acentuada. Dessa forma alguns trabalhos fazem segmentação da imagem utilizando a variância de uma determinada região na imagem[10]. Outras formas de segmentar são utilizadas, mas optou-se por utilizar a variância por ser mais simples e também eficaz.

A variância é uma medida estatística que consiste no desvio quadrático médio da medida de um conjunto de dados, sua fórmula é mostrada na equação 3.1. No caso de imagens o cálculo da variância é feito utilizando os valores dos *pixels* de uma região na imagem. Após o cálculo, o valor da variância é comparado com um limiar de classificação pré-estabelecido. Caso a variância seja menor que o limiar, esta é classificada como fundo, se a variância supera este limiar a região é classificada como área de interesse.

$$s = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1} \quad (3.1)$$

O tamanho da região para o cálculo da variância, e o limiar de classificação devem ser estabelecidos previamente mediante a testes de forma empírica. Deve-se estabelecer esses parâmetros para que se adéquem a cada tipo de imagem de impressão digital, sabendo que a coleta feita com tecnologias distintas geram imagens com características diferentes.

3.2. Filtro de Gabor

Após definido a região de interesse com a segmentação é preciso melhorar a qualidade das cristas da impressão digital. Estas apresentam algumas imperfeições que são peculiares das digitais, ou ocorrem devido a tecnologia de coleta empregada. Em

alguns casos se a imagem possuir uma qualidade boa, são empregados filtros gaussianos e filtros passa-altas como visto em [2]. Mas no caso de imagens mais ruidosas é preciso de um filtro mais robusto. Nesse trabalho foi utilizado o filtro de Gabor, como sugerido por [5,9].

Esse filtro foi proposto inicialmente por Dennis Gabor em 1945 [4]. É muito utilizado para a melhoria da qualidade de impressões digitais e em processos de análise de textura[5,10]. A formulação do filtro é vista na equação 3.2.

$$G(x, y, f) = \exp \left[-0.5 \cdot \left(\frac{x_r^2}{gx^2} + \frac{y_r^2}{gy^2} \right) \right] \cdot \cos(2\pi \cdot f \cdot y_r)$$

onde

$$\begin{aligned} x_r &= x \cdot \cos(\theta) - y \cdot \sin(\theta) \\ y_r &= y \cdot \cos(\theta) + x \cdot \sin(\theta) \end{aligned} \quad (3.2)$$

O filtro consiste em um envelope gaussiano com uma função cosseno. O seu resultado é uma suavização da imagem, onde também ocorre destaque das senóides com uma determinada freqüência[8]. No caso de aplicá-lo em impressões digitais o filtro destaca as senóides representadas pelas cristas e os vales e suprime os ruídos.

Para um bom funcionamento do filtro é preciso estabelecer adequadamente o tamanho da região onde este será aplicado. O nível de suavização pode ser calibrado nos parâmetros *gx* e *gy*. O parâmetro da freqüência *f* pode ser estimado, baseado em medições das distâncias entre os topos das cristas das digitais. Já o parâmetro de direção das senóides *θ* precisa ser obtido previamente através de uma mapa de direções.

3.3. Mapa de direções

Para obter o mapa de direções foi utilizado um método baseado no filtro de detecção de bordas de Sobel. Aplicou-se os filtros de detecção de bordas utilizando a convolução com máscara que detectam bordas nas direções de 0°, 45°, 90°, e 135°. Em seguida, verificou-se a direção que obteve maior estímulo aos filtros e essa direção foi selecionada para aquela região, ou seja é escolhido a direção representada pela máscara que detectou mais bordas.

Por causa das imperfeições das imagens, somente com essa etapa não foi possível gerar um mapa correto de direções para o filtro de Gabor. Em algumas regiões

a direção detectada não foi a correta, mas observou-se que na maior parte dos casos a direção detectada foi a correta. Então foi proposto após essa etapa fazer uma uniformização do mapa de direções, aplicando um filtro que utiliza a moda. Para uma região central foi atribuída a direção que aparece mais vezes nas suas oito regiões vizinhas, ou seja o valor da moda da vizinhança.

Na figura 3.1 é visto o mapa de direção sem a uniformização e a aplicação do filtro de Gabor com esse mapa, em seguida é visto o mapa com a uniformização e o seu resultado com o filtro de Gabor. No mapa as regiões mais escuras representam 0° graus e as mais claras representam 45°, 90°, 135° gradativamente.

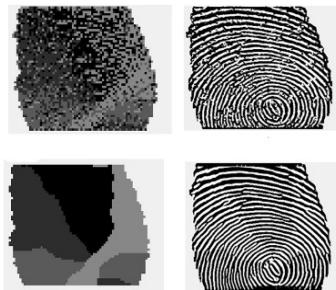


Figura 3.1 imagem do método de detecção da direção o filtro de Gabor, em baixo a detecção da direção com a uniformização e o resultado do filtro de Gabor.

3.4. Afinamento

Após obter uma imagem com boa qualidade, foi utilizado um método de afinamento para reduzir a representação das cristas da impressão digital para um esqueleto digital. Essa redução gera uma imagem com o mesmo conteúdo mas com cristas de um *pixel* de espessura. Foi utilizado o filtro de Zhang e Suen para fazer o afinamento da imagem [6].

3.5. Extração e eliminação de falsas minúcias

No esqueleto digital da imagem é preciso localizar as minúcias. Para tal foi utilizado o algoritmo *cross-number*[2], que verifica as transições de cor dos oito vizinhos de um *pixel*. Para um *pixel* na imagem ser considerado uma minúcia do tipo terminação este deve ter uma única transição de cor e do tipo bifurcação três transições.

Nem todas as minúcias encontradas foram verdadeiras. O algoritmo detectou falsas terminações nas extremidades do esqueleto digital, estas foram eliminadas. Minúcias muito próximas também foram removidas onde porque caracterizavam cristas fragmentadas.

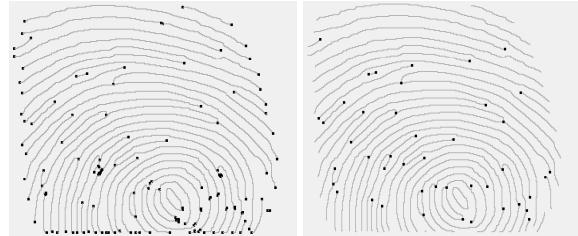


Figura 3.2 minúcias detectadas sem eliminação das minúcias falsas e em seguida com a eliminação.

4. Testes

Para testar os métodos selecionados, foram utilizadas imagens capturadas por um leitor Digital Persona e também imagens da base de dados da FVC de 2004 [3], na qual consiste em um banco de dados com 4 tipos de imagens de impressão digital, coletadas com diferentes tecnologias, sendo que a quarta base dados são imagens sintéticas geradas por *software*. A FVC (*Fingerprint Verification Competition*) é uma competição de softwares de reconhecimento de impressão digital, organizado por algumas universidades internacionais. Na figura 4.1 é mostrado as imagens utilizadas para os testes.

Na etapa de segmentação foi utilizado uma região com janela de tamanho 13x13 porque foi a que melhor se adequou as imagens de resolução de 500dpi, o limiar da variância variou entre 500 e 3000. Para o cálculo da direção foram utilizadas janelas de 5x5 e quatro direções: 0°, 45°, 90° e 135°, sendo que para a aplicação do filtro essas direções foram convertidas para radianos. O filtro de Gabor foi aplicado com uma janela de 13x13 e com parâmetro da freqüência estabelecida em 0.7. Este valor representa o inverso do dobro da largura das cristas [8].

A comparação dos resultados gerados pelos algoritmos proposto com os resultados esperados, foi feita por uma avaliação subjetiva dos autores. Esperasse investigar posteriormente um critério automatizado para avaliar a extração de minúcias.

Imagens originadas do leitor Digital Persona obtiveram os melhores resultados, tanto na segmentação quanto no filtro de Gabor e detecção das minúcias.

Nas imagens do primeiro e terceiro banco de dados da FVC os resultados foram satisfatórios. Destaca-se o resultado da segmentação no terceiro banco de dados da FVC (terceira imagem da figura 4.1) onde a segmentação foi eficiente mesmo com um fundo imagem com variação de cor mais acentuadas do que as imagens dos outros banco de dados.

No segundo banco de dados da FVC a segmentação em algumas imagens não foi eficiente, devido ao baixo contraste com o fundo. No quarto banco de dados da FVC, que são imagens sintéticas produzidas por *software*, o mapa de direção gerado em algumas imagens não foi o esperado, devido as cristas das digitais estarem muito fragmentadas.



Figura 4.1 – Exemplo de imagens do banco de dados da FVC de 2004 e uma imagem coletada por um leitor Digital Persona.

5. Conclusões e trabalhos futuros

Foi constatado a eficiência do método de segmentação que utiliza a variância. O filtro de Gabor também mostrou bons resultados mas este é extremamente dependente da etapa de detecção da direção das cristas da impressão digital. Os resultados das outras etapas são originados do sucesso dessas duas etapas iniciais.

No geral pode-se considerar satisfatório o conjunto dos métodos selecionados, principalmente por causa de ótimos resultados obtidos com as imagens capturadas pelo leitor Digital Persona. Entretanto a utilização do banco de dados da FVC foi importante para verificar o comportamento dos algoritmos com imagens degradadas e geradas por tecnologias diferentes.

Para os trabalhos futuros pretende-se estender as pesquisas buscando soluções para os casos onde a metodologia proposta não foi eficiente. A continuidade do trabalho dar-se-á com a investigação e

seleção de técnicas de comparação de minúcias. Posteriormente pretende-se fazer uma estruturação para utilizar esse sistema de reconhecimento de impressão digital para a autenticação de usuários na plataforma *Web*.

6. Referências

- [1] ARAÚJO, Marcos Elias Cláudio de, PASQUALI, Luiz, HISTÓRICO DOS PROCESSOS DE IDENTIFICAÇÃO. Disponível em: <http://www.papiloscopistas.org/historico.doc>
- [2] FARIA, Diego Resende, Reconhecimento de Impressões Digitais com Baixo Custo Computacional para um Sistema de Controle de Acesso, Dissertação (mestrado). UFPR, Curitiba, 2005.
- [3] Fingerprint Verification Competition. Disponível em: <http://bias.csr.unibo.it/fvc2006/>
- [4] Gabor, D., "Theory of Communication", Journal of the Institute of Electrical Engineers, 1946, pp. 429-457
- [5] GONZAGA, S. L. de O. ; VIOLA, F. ; CONCI, A. . Filtro adaptativo para melhoria de imagens de impressões digitais utilizando o filtro de Gabor e campos direcionais. In: 4º Congresso Temático de Dinâmica, Controle e Aplicações, 2005, Bauru.
- [6] GONZALEZ, Rafael C., WOODS, Richard E., Processamento de Imagens de Impressões Digitais. São Paulo: Edgar Blucher, 2000.
- [7] Jain, A. K, and Maltoni, D., "Handbook of Fingerprint Recognition", Springer, New York, 2003.
- [8] MARQUES, Airam Carlos Pais Barreto, Extração de Minúcias em Imagens de Impressões Digitais utilizando Redes Neurais, Dissertação(mestrado) NCE-UFRJ, Rio de Janeiro, 2004.
- [9] OLIVEIRA, Sanderson Lincoln Gonzaga de, Desenvolvimento de um algoritmo baseado no filtro de Gabor para identificação de impressões digitais. Dissertação(mestrado), UERJ,Nova Friburgo, 2004.
- [10] SÁ, Gustavo Ferreira Cardoso de, Melhorias no Reconhecimento de Impressões Digitais Baseado no Método FingerCode. Dissertação(mestrado), UNICAMP, Campinas, 2006.

Extensão do Algoritmo CCV Aplicado à Recuperação de Imagens com Base no Conteúdo Visual

Rodrigo Silva Oliveira¹, Thatyene Louise Alves de Souza Ramos¹,
Ana Paula Brandão Lopes (Co-orientadora)^{1,2}, Camillo Jorge Santos Oliveira (Co-orientador)^{1,3},
Arnaldo de Albuquerque Araújo (Orientador)¹

¹Departamento de Ciência da Computação - Universidade Federal de Minas Gerais

²Departamento de Ciências Exatas e Tecnológicas - Universidade Estadual de Santa Cruz

³Departamento de Informática - Pontifícia Universidade Católica de Minas Gerais

{rsilva, thatyene, paula, camillo, arnaldo}@dcc.ufmg.br

Resumo

Este trabalho propõe a implementação de uma extensão do algoritmo do *Color Coherence Vectors* (CCV) que incorpora mais informações ao atributo cor das imagens, denominando-se *Extended CCV*. O *ECCV* leva em consideração a localização dos componentes conectados na imagem, bem como a sua aplicação (utilização) online. Aplicou-se o *ECCV* à base de imagens da Pesquisa Guignard, como ferramenta de Recuperação de Informação com Base no Conteúdo (RIBC).

1. Introdução

Neste trabalho, objetiva-se descrever a implementação de uma ferramenta de recuperação de imagens com base no conteúdo visual aplicado à Pesquisa Guignard. A Pesquisa Guignard¹ é um projeto multidisciplinar iniciado em 2002 na escola de Belas Artes da UFMG com a intenção de organizar um amplo acervo sobre Alberto da Veiga Guignard (1896 - 1962), renomado pintor da moderna arte brasileira, assim como realizar aprofundado estudo de sua obra.

Segundo Del Bimbo [2], recuperação de informação visual é um assunto amplamente pesquisado na tecnologia de informação. O propósito geral das aplicações dessa área é recuperar, de um banco de dados, imagens ou seqüências de imagens que são relevantes para uma consulta. É, portanto, uma extensão da recuperação de informação tradicional, projetada para incluir mídias visuais.

Muitos métodos foram propostos para possibilitar a recuperação de imagens por meio de atributos de cor. Um desses métodos é o *Color Coherence Vectors* (CCV), proposto por Pass [3], que consiste em agregar informação espacial ao tradicional histograma de cores. Apesar de ser uma técnica simples, ela apresenta bons resultados e grande eficiência em termos de custo computacional, sendo possível utilizá-la em sistemas interativos de recuperação de informação visual na Internet.

Um problema observado no algoritmo CCV original é que a proposta não leva em consideração a localização dos conjuntos de pixels que formam os componentes conectados da imagem. A Figura 1 mostra esse problema:



Figura 1. No CCV original, as duas figuras acima são representadas da mesma maneira.

Uma imagem composta de uma cor na metade esquerda e outra cor na metade direita tem o mesmo CCV de uma imagem com as mesmas cores dispostas em lados contrários, como mostrado na figura anterior. O algoritmo CCV classificaria as duas imagens como sendo idênticas. Entretanto, essas duas imagens são diferentes para um observador, devido à localização das cores.

Este artigo apresenta o sistema RIBC desenvolvido e aplicado à Pesquisa Guignard, além de propor uma extensão para a técnica do CCV, denominada *ECCV* (*Extended Color*

1 Site da pesquisa: <http://www.pesquisaguignard.eba.ufmg.br>

Coherence Vectors), onde é considerada a localização dos componentes conectados na imagem. Além disso, é discutida a utilização da técnica em uma ferramenta *online* de recuperação de imagens aplicada à base de obras da Pesquisa Guignard.

2. O sistema RIBC

Neste projeto, foi desenvolvida uma interface *WEB* onde o usuário pode escolher uma obra como imagem-consulta e acionar uma pesquisa por obras semelhantes, onde é utilizado o algoritmo ECCV para comparação das obras e ordenação do resultado. O diagrama da Figura 2 mostra o funcionamento do sistema:

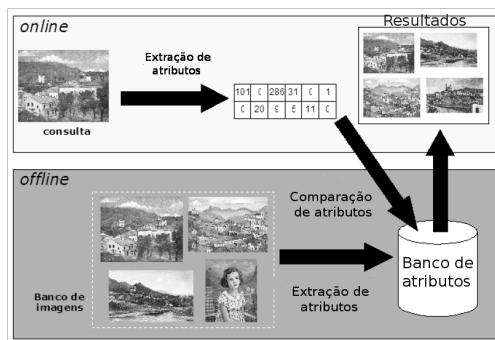


Figura 2. Funcionamento da ferramenta de recuperação de imagens.

Em uma etapa *offline*, todas as imagens da base são processadas e têm seus atributos extraídos e armazenados em uma base de dados. Na etapa *online*, o usuário seleciona uma imagem, seus atributos são extraídos e comparados com os atributos de todas as obras que compõem a base. O resultado é dado em termos da exibição das imagens em ordem decrescente de similaridade com a imagem de consulta.

A Figura 3 apresenta a tela com o resultado de uma consulta.

3. Extended CCV (ECCV)

No CCV, é realizada a quantização das cores presentes na imagem e, a partir do cálculo de componentes conectados, são criados dois vetores. O vetor de pixels coerentes, que armazena o percentual de pixels da mesma cor que encontram-se agrupados em componentes com um número de pixels acima de um determinado limiar. E o vetor de pixels incoerentes, que armazena o percentual de pixels da mesma cor que encontram-se agrupados em compo-



Figura 3. Interface WEB para o sistema RIBC.

nentes com um número de pixels abaixo desse limiar. Esses dois vetores são usados como vetores de características para indexação das imagens.

O ECCV estende o CCV para adicionar alguma informação sobre a localização dos pixels. O processo de extração do ECCV é dividido em quatro etapas, detalhadas a seguir.

3.1. Quantização de cores

A primeira etapa da extração de atributos é feita quantizando-se as cores presentes na imagem. Essa etapa é necessária para reduzir o tamanho dos vetores de atributos a serem extraídos. Neste trabalho, as cores das imagens são convertidas para o espaço de cores IHS, que representa melhor a percepção de cores dos seres humanos. Apenas o componente H (*hue* ou matiz) é utilizado no processo de quantização. Testes empíricos mostraram que 16 faixas de quantização são suficientes para obter-se bons resultados na recuperação.

3.2. Cálculo de componentes conectados

A rotulação de componentes conectados consiste na varredura da imagem e agrupamento de pixels da mesma cor que sejam vizinhos uns dos outros em uma mesma região. As informações armazenadas para cada região são a cor dos pixels e o número de pixels que compõem essa região.

3.3. Cálculo do ECCV

No ECCV, a imagem é dividida em partes de mesmo tamanho (como pode ser visto na Figura 4) e, para cada parte, o vetor de assinatura CCV é calculado da seguinte maneira:

Cada conjunto de pixels de uma determinada cor do histograma é particionado em dois tipos: coerente, se pertence

a uma região grande (contendo pelo menos 1% dos pixels da parte da imagem) e uniformemente colorida, e incoerente, se não pertence a nenhuma dessas regiões. Assim, se α_i denota os *pixels* coerentes da i -ésima cor das 16 utilizadas e β_i os *pixels* incoerentes, então o vetor $\langle(\alpha_1, \beta_1), (\alpha_2, \beta_2), (\alpha_3, \beta_3), \dots, (\alpha_N, \beta_N)\rangle$ corresponde à assinatura da parte da imagem.

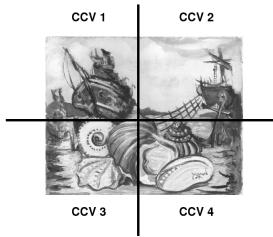


Figura 4. Divisão da imagem em partes e cálculo do CCV para cada uma dessas partes.

3.4. Comparação de imagens usando ECCV

A comparação das imagens é feita *online*.

As imagens são comparadas por intermédio da Equação 1.

$$\Delta = \frac{\sum_{i=1}^4 \delta_i}{4} \quad (1)$$

onde δ_i é calculado com relação às partes da imagem na mesma posição por intermédio da Equação 2.

$$\delta_i = \sum_{j=1}^n \left| (\alpha_j - \hat{\alpha}_j) + (\beta_j - \hat{\beta}_j) \right|, \quad i \in \{1, 2, 3, 4\} \quad (2)$$

que é a soma da distância L_1 (soma dos valores absolutos das diferenças) entre os vetores coerentes e incoerentes das duas imagens sendo comparadas, segundo Pass [3].

Dessa forma, quanto maior o valor de Δ , menos similaridade existe entre as imagens.

Na situação em que a imagem-exemplo tenha o mesmo valor de similaridade para duas ou mais imagens diferentes, o desvio padrão σ (Equação 3) das distâncias dos quadrantes é utilizado como critério de desempate. Nesse caso, quanto menor o valor de σ , maior a similaridade entre as imagens.

$$\sigma = \sqrt{\frac{1}{4} \sum_{i=1}^4 (\delta_i - \Delta)^2} \quad (3)$$

4. Resultados e discussões

Para a realização dos experimentos, foram utilizadas 132 imagens, com resolução de 100 dpi e dimensões variáveis de no máximo 800 x 800 pixels, selecionadas da base da Pesquisa Guignard. Os processos de extração dos atributos e recuperação de imagens foram realizados em uma máquina com processador de 3.0 GHz e 1 GB de memória RAM.

A primeira parte dos experimentos consistiu em avaliar o tempo gasto para cálculo do ECCV variando-se o número de cores utilizadas na quantização (2, 4, 8, ..., 256), mantendo-se o número de partes constante e igual a 1 (imagem sem ser dividida). O tempo de extração dos atributos variou entre 2 e 2,5 segundos para cada imagem. Em um segundo experimento, variou-se o número de divisões feitas na imagem (1, 2, 3, ..., 10) e observou-se o impacto das mesmas no tempo gasto para a extração dos atributos. Mais uma vez o tempo gasto para cada imagem ficou entre 2 e 2,5 segundos. Como a extração dos atributos é realizado em uma fase *offline*, o tempo observado não traz problemas ao sistema.

O gráfico da figura 5 ilustra o resultado obtido na apuração do tempo médio gasto pelo algoritmo para fazer a comparação entre duas imagens quando se varia o número de cores utilizadas na quantização. Apesar de o tempo de resposta crescer linearmente com o aumento do número de cores, observa-se que os valores são baixos, não chegando a 0,04 segundos quando se utiliza 256 cores. Para uma consulta na ferramenta implementada e utilizando a base de 132 imagens, o tempo de resposta não chega a 4 segundos, o que seria aceitável para o propósito da ferramenta. Entretanto, decidiu-se utilizar 16 cores na quantização, já que a recuperação não parece ser afetada significativamente enquanto que o tempo total de resposta passa a ser de aproximadamente 3 segundos.

Já o gráfico da figura 6 mostra a relação entre a divisão da imagem para o cálculo do CCV de cada parte e o tempo médio gasto para a comparação entre duas imagens. É possível observar nesse gráfico que a divisão da imagem faz com que o tempo médio para comparação aumente consideravelmente, chegando a aproximadamente 0,43 segundos para a divisão em 10 partes. Tal fato impõe uma limitação clara no número de partes utilizadas. Para a utilização de 10 partes, o sistema gasta em torno de 56 segundos para exibir a resposta, o que é intolerável para uma ferramenta interativa na *WEB*. Para equilibrar o compromisso entre a qualidade da resposta e o tempo gasto para gerá-la, o algoritmo está parametrizado para dividir as imagens em 4 partes, gastando em média 10 segundos para gerar a resposta.

Vale ressaltar que os tempos foram obtidos em uma máquina *desktop* comum, e podem ser bem menores em um servidor com mais recursos.

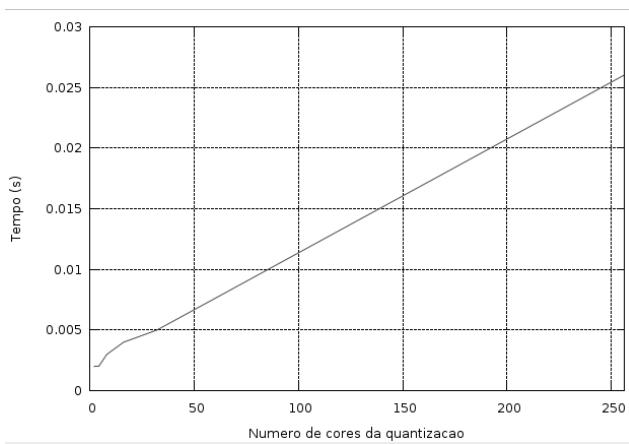


Figura 5. Teste para comparação de atributos variando-se o número de cores da quantização.

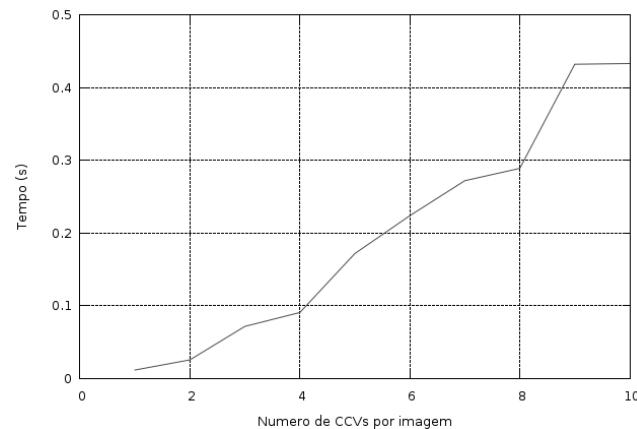


Figura 6. Teste para comparação de atributos variando-se o número de partes da imagem.

A Figura 7 mostra o resultado de quatro consultas. A primeira imagem de cada linha é a imagem pesquisada e as subsequentes são as cinco primeiras retornadas pelo sistema em ordem decrescente de similaridade.

5. Conclusões e trabalhos futuros

Este trabalho foi motivado pela necessidade de implementar um sistema de busca de imagens que oferecesse maior flexibilidade do que os sistemas de busca baseados em informações textuais. Os resultados apresentados pelo sistema são satisfatórios, tanto em ter-



Figura 7. Resultado de algumas consultas efetuadas no sistema RIBC desenvolvido.

mos do tempo de resposta para uma consulta quanto em termos da qualidade da resposta. O sistema está sendo utilizado experimentalmente por meio do endereço <http://www.pesquisaguignard.eba.ufmg.br/SearchByExample>.

Como trabalhos futuros, pretende-se agregar outros algoritmos de recuperação de imagens com base no conteúdo ao sistema RIBC implementado, oferecendo aos usuários uma ferramenta flexível e rápida para a realização de consultas de obras de Guignard.

Os autores são gratos à CAPES, CNPq e FAPEMIG pelo suporte financeiro.

Referências

- [1] M. C. R. Barbosa. Protótipo para recuperação de imagens com base no conteúdo aplicado à pesquisa guignard. Projeto Orientado de Computação II - DCC/UFMG.
- [2] A. Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann, San Francisco, CA - USA, 1999.
- [3] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *Proceedings of the fourth ACM international conference on Multimedia*, pages 65–73, 1997.
- [4] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE transactions Pattern Analysis Machine Intelligence*, 22 - 12:1349 – 1380, 2000.

Estruturas Tritree

Compressão de imagens coloridas

Tiago Oliveira Cunha¹, Vânia Cordeiro da Silva (orientadora)¹

¹Departamento de Ciências Exatas e Tecnológicas -DCET

Universidade Estadual de Santa Cruz (UESC) – Ilhéus – BA – Brazil

tiago_cic@yahoo.com.br, vania@uesc.br

Resumo.

Um novo método para segmentação/decomposição aplicada à compressão de imagens foi criado [5], o algoritmo de decomposição Tritree (TT). O objetivo é a segmentação da imagem em um conjunto de regiões triangulares com cores homogêneas, onde a diferença entre os valores de pixels não ultrapassem um determinado limite. Uma árvore é construída para representar a decomposição. Cada triângulo será um nó da TT. O triângulo inicial contendo toda a imagem é a raiz. Os últimos triângulos, que representam a imagem comprimida, são as folhas (nós terminais). Resultados experimentais com imagens em tons de cinza mostram que o método proposto possui resultados melhores do que a decomposição Quadtree (QT), em termos de PSNR e a taxa de compressão. É testado agora, o método desenvolvido para imagens coloridas.

1. Introdução

Muitas técnicas de compressão de imagens são baseadas em segmentação ou decomposição da imagem original em regiões homogêneas, o que pode ser representado por um reduzido número de bits. Este é o caso do conhecido método de decomposição *Quadtree* (QT), que tem sido empregado em diversos métodos híbridos de compressão de imagens encontrados na literatura [3, 4, 6, 7].

Este artigo apresenta a técnica de decomposição *Tritree* (TT), para representação de imagens reais bidimensionais, em tons de cinza, em diferentes níveis de resolução, desenvolvido em [5], e estende-a para imagens em cores espectrais. A decomposição TT é similar à decomposição QT. Contudo, enquanto QT trabalha subdividindo a imagem em regiões quadráticas progressivamente menores, a decomposição TT o faz para regiões triangulares. A meta é segmentar a imagem em um conjunto de regiões triangulares homogêneas, isto é, regiões onde a diferença dos valores de nível de cor dos *pixels* não excede um determinado limiar. A estrutura de dados árvore de pesquisa é empregada para representar a decomposição. Cada elemento, ou triângulo, forma um nodo da árvore. O elemento inicial, que contém a imagem completa, é a raiz da árvore e suas folhas (nós terminais) guardam a

informação relevante da imagem comprimida, ou seja, a informação necessária para sua descompressão. Todas as árvores foram construídas por um procedimento *top-down*.

A aplicação mais imediata para este método seria em algoritmos híbridos de compressão de imagens. Entretanto, várias outras aplicações que fazem uso de decomposição ou segmentação, também seriam viáveis.

2. Imagens Coloridas

Em uma imagem digital colorida no sistema RGB, pode-se considerar um pixel como um vetor $f(i,j) = (fr(i,j), fg(i,j), fb(i,j))$, cujas componentes representam as intensidades de vermelho, verde e azul da cor. Pode-se considerar, portanto, que uma imagem colorida é a composição de três imagens monocromáticas, $fr(i,j)$, $fg(i,j)$, $fb(i,j)$, denominadas, respectivamente, de *banda vermelha* (ou *banda R*), *banda verde* (ou *banda G*), e *banda azul* (ou *banda B*) da imagem colorida, conforme ilustra a Figura 1.

O modelo RGB é o mais utilizado para representar cores em hardwares e aplicações que a suportam. É o modelo empregado nos dispositivos de visualização gráficos [2].



Figura 1. Imagem colorida e suas bandas

3. Decomposição Tritree

Faz-se necessário algumas definições, observando o armazenamento das informações na estrutura de dados árvore. Cada item de dado armazenado numa árvore é chamado de nodo. O nodo inicial é a raiz. Sub-árvores podem originar de cada nodo, estabelecendo uma relação de pai-filho entre os nodos. Um nodo que não possui filhos é chamado nodo terminal ou folha. O número de filhos de um nodo determina seu grau. O número de gerações entre um nodo e a raiz da árvore determina o nível do nodo.

Todo o processo é realizado em cada uma das 3 bandas da imagem colorida. Decomposição TT começa pelo enquadramento da imagem em um triângulo equilátero, que é sempre sub-dividido em quatro novos triângulos equiláteros. O grau de homogeneidade é calculado para cada triângulo gerado e testado contra um valor de limiar dado empiricamente pelo usuário, para determinar se este deve ser novamente dividido ou não. Este processo é repetido um número fixo de vezes ou até todos os triângulos testados serem rotulados como homogêneos. Cada triângulo será um nodo da TT. O triângulo inicial, que contém a imagem toda, será a raiz. Os triângulos finais, representando a imagem decomposta, são as folhas.

A Figura 2 exemplifica a decomposição TT de uma imagem qualquer, resultando em uma árvore de grau 4. Uma TT com n níveis pode ter até 4^{n-1} folhas. A Figura 3 mostra como as informações dos nodos da TT são armazenadas, utilizando um registro com 9 campos para cada nodo.

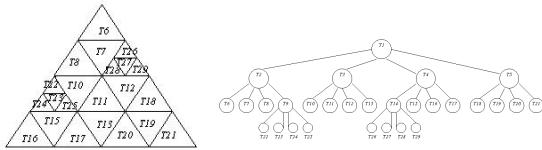


Figura 2: Informações de uma Estrutura Tritree

1	1	p1	p2	p3	2	3	4	5	0
2	-2	p1	p4	p5	0	0	3	0	1
3	-2	p4	p5	p6	2	4	5	0	1
4	-2	p4	p2	p6	0	3	0	0	1
5	-2	p5	p6	p3	3	0	0	0	1

Figura 3: Informações em uma Estrutura Tritree

A lista dos vértices dos triângulos é organizada em uma árvore binária (grau 2). Registros com cinco campos são necessários para armazenar as informações de cada nodo da árvore binária: dois

campos para as coordenadas X-Y dos vértices, dois outros campos apontando para os vértices filhos, e o último guarda a informação do valor da cor do vértice. Quando um triângulo é dividido novos nodos são adicionados na árvore de vértices (a árvore binária). Estes nodos são os pontos do meio de cada aresta do triângulo, e correspondem aos vértices dos quatro novos triângulos criados pela decomposição. Os pontos são inseridos na árvore binária em ordem crescente, de acordo com os valores X-Y de suas coordenadas. Um ponto com coordenada (x,y) é definido como sendo menor do que outro ponto com coordenadas (u,v), respectivamente, se:

$$\begin{cases} x < u \text{ ou} \\ x = y \text{ e } y < v \end{cases}$$

A Figura 4 apresenta como a imagem é enquadradada no triângulo inicial. A largura e altura do triângulo são determinadas de acordo com as dimensões da imagem, garantindo, assim, o enquadramento perfeito. A altura a do triângulo raiz é dado por: $a = l + c * \text{sen}60^\circ$

Onde l é o número de linhas da imagem e c é o número de colunas. A largura b do triângulo é calculada como se segue: $b = c + 2 * l * \text{cos}60^\circ / \text{sen}60^\circ$ Em geral, a e b são valores fracionários, requerendo aproximação para o mais próximo, um dos valores da malha de coordenadas da imagem discreta. Como consequência, os cantos superiores da imagem, muito provavelmente, não tangencie as arestas do triângulo inicial como ilustrado na Figura 4.

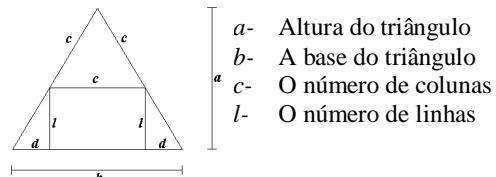


Figura 4: Imagem Enquadradada em Triângulo Inicial

Internamente, a estrutura de dados utilizada para representar imagens é uma matriz bi-dimensional [4].

4. Reconstrução

A reconstrução da imagem é realizada através da reconstrução dos triângulos produzidos pela decomposição TT. Dois diferentes algoritmos de reconstrução foram pesquisados: replicação da média e interpolação linear [3]. O primeiro é representado pela média dos valores de níveis de cor do triângulo, a reconstrução consiste da replicação deste valor médio dentro dos limites físicos dos triângulos. A interpolação linear é executada em cada triângulo-folha (nodo terminal) da TT, de acordo com a equação: $ax + by + cz = d$.

Abaixo, na Figura 5, as imagens reconstruídas utilizando o método Tritree com a reconstrução feita por interpolação linear.



Figura 5: Imagens reconstruídas

5. Codificação

O procedimento de codificação inclui codificar a informação da estrutura da árvore, bem como codificar os níveis de cor dos vértices dos triângulos. Para a codificação da estrutura da árvore o valor '1' é utilizado para rotular os nodos pais e '0' é reservado para rotular os nodos folhas [5][6]. Todos os nodos do último nível de decomposição são folhas, portanto, o processo de codificação da estrutura da árvore pode parar no penúltimo nível. O código da raiz, único nodo do primeiro nível, será sempre '1', podendo ser suprimida, também, esta informação. Por exemplo, o código resultante para a árvore TT apresentada na Figura 2 é 1111 - 0001 - 0000 - 1000 - 0000.

A informação do nível de cor de uma imagem reconstruída por interpolação linear entre os vértices dos triângulos é armazenada utilizando-se codificação preditiva sem perdas. Somente os vértices dos nodos folhas são armazenados. Um ponto de vértice, o primeiro a ser calculado, é escolhido para ser o preditor e o único a ser codificado com 8 bits (para o caso de imagens com 256 níveis de cor). Para os outros pontos de vértice, somente a diferença entre estes e o preditor é codificada, com o mesmo número de bits do limiar. Isto acontece por que triângulos adjacentes tem, no mínimo, um vértice em comum [2][3].

6. Teste de Homogeneidade

Foram utilizados dois testes de homogeneidade, no primeiro, descobrem-se os pixels de maior e menor valor em seguida a diferença entre os dois é com parada com o limiar de decisão informado anteriormente (foram utilizados dois tipos de limiares, o primeiro é passado de forma empírica, o

segundo utiliza um valor passado empiricamente para calcular um novo valor de limiar para cada nível da arvore TT é utilizado em conjunto com o teste de homogeneidade desvio da média), caso seja maior há divisão, senão o triângulo é considerado terminal, o segundo é o desvio da média [7], a decisão ou não da divisão é feita através da comparação do valor absoluto da diferença dos valores de níveis de cor de cada pixel do triângulo com o valor médio dos níveis de cor dos pixels do triângulo, essa diferença é comparada com o valor do limiar de divisão caso seja maior há a divisão.

7. Medidas de Avaliação

A eficiência da compressão é medida pela taxa de compressão, expressa como a média (Tx) do número de bits requeridos por pixel para codificar a imagem comprimida, dado por:

$$Tx \text{ bpp} = \frac{\text{Número de bits para codificar a imagem comprimida}}{\text{Total de pixels na imagem original}}$$

A qualidade da imagem reconstruída é mensurada pela relação sinal-ruído de pico, ou PSNR, do inglês “peak signal-to-noise ratio”, calculada como:

$$PSNR = 10 \cdot \log_{10} \left\{ \frac{A^2}{MSE} \right\} dB$$

Nesta equação, A representa o valor máximo de nível de cor que os pixels da imagem testada podem alcançar. Nos testes realizados foram utilizadas apenas imagens com 8 bits/pixel, ou seja, o valor de A foi sempre igual a 255. A variável MSE é o erro médio quadrático (Mean Square Error) da diferença, ou erro, entre o sinal da imagem original ($I(i,j)$) e a imagem comprimida ($R(i,j)$). Sendo L o número de linhas das imagens e C o número de colunas, o MSE é calculado como se segue:

$$MSE = \frac{1}{LC} \left\{ \sum_{i=0}^L \sum_{j=0}^C [I(i,j) - R(i,j)]^2 \right\}$$

8. Resultados Experimentais

Serão apresentados aqui os resultados relativos à imagem Lena por se tratar da imagem mais utilizada em testes de algoritmos de compressão de imagens.

Nos testes foram utilizados os métodos Tritree e Quadtree, sendo que no primeiro foram testadas duas estratégias de reconstrução, replicação da média das cores e interpolação linear das cores e dois testes de homogeneidade, diferença entre maior e menor pixel e desvio da média, no segundo somente a replicação da média e diferença entre maior e menor pixel.

Os cálculos foram feitos para as bandas da imagem logo cada medida avaliativa vem acompanhada da banda a qual faz referência, também foram calculados o número de vértices e o

número partições (triângulos ou quadrado, de acordo com o método) gerados por cada método, os resultados obtidos para o método Tritree estão expostos nas Tabelas 1 (reconstrução por replicação da média), Tabela 2 (reconstrução por interpolação linear) e Tabela 4 (reconstrução por replicação e desvio da média), os resultados referentes ao método Quadtree são expostos na Tabela 3.

Tabela 1:Tritree: reconstrução por replicação da média.

Limiar	Vértices	Partições	MSE(R)	MSE(G)	MSE(B)	PSNR(R)	PSNR(G)	PSNR(B)
5	188028	415645	106,00	105,00	80,00	27,8777	27,9189	29,0999
10	129612	277079	153,00	121,00	94,00	26,2838	27,3029	28,3995
20	78109	159949	200,00	153,00	125,00	25,1205	26,2838	27,1617
30	55921	111527	234,00	184,00	152,00	24,4386	25,4826	26,3123
50	33989	66106	307,00	253,00	215,00	23,2594	24,0995	24,8064

Tabela 2:Tritree: reconstrução por interpolação linear.

Limiar	Vértices	Partições	MSE(R)	MSE(G)	MSE(B)	PSNR(R)	PSNR(G)	PSNR(B)
5	188028	415645	59,00	43,00	41,00	30,4222	31,7961	32,0029
10	129612	277079	107,00	59,00	55,00	27,8369	30,4222	30,7271
20	78109	159949	154,00	92,00	88,00	26,2555	28,4929	28,6859
30	55921	111527	191,00	123,00	116,00	25,3204	27,2317	27,4862
50	33989	66106	264,00	197,00	180,00	23,9147	25,1861	25,5780

Tabela 3:Quadtree original.

Limiar	Vértices	Partições	MSE(R)	MSE(G)	MSE(B)	PSNR(R)	PSNR(G)	PSNR(B)
5	555928	890084	180,00	107,00	90,00	25,5780	27,8369	28,5883
10	353523	546892	184,00	111,00	94,00	25,4826	27,6775	28,3995
20	191293	282384	197,000	125,00	106,00	25,1861	27,1617	27,8777
30	126421	180024	216,00	144,00	125,00	24,7862	26,5471	27,1617
50	66384	90676	272,00	194,00	181,00	23,7851	25,2527	25,5540

Tabela 4:Tritree: reconstrução por interpolação linear, desvio da média.

Limiar	Vértices	Partições	MSE(R)	MSE(G)	MSE(B)	PSNR(R)	PSNR(G)	PSNR(B)
5	270222	508360	83,00	48,00	48,00	28,9400	31,3183	31,3183
10	147478	278422	131,00	76,00	73,00	26,9580	29,3226	29,4975
20	72849	137042	188,00	135,00	116,00	25,3892	26,8274	27,4862
30	45733	86136	234,00	185,00	165,00	24,4386	25,4590	25,9559
50	23125	44862	344,00	296,00	273,00	22,7652	23,4178	23,7691

Observando-se a coluna vértices e partições das tabelas, pode-se observar que o número de vértices e partições gerados pela Quadtree é sempre maior que o número de vértices e partições gerado pela Tritree. Este fato é demonstrado matematicamente por Mayer em [6], que conclui no final de sua pesquisa: "Dada uma imagem I , particionada em um dado número de regiões de mesma área A , o particionamento triangular requer cerca de metade dos vértices requeridos para o particionamento em quadrados". Para o particionamento em regiões de diferentes áreas, esta afirmação pode ser verificada empiricamente através dos experimentos, tanto neste trabalho como no de Mayer [6]. Isto implica em uma quantidade menor de informação passada ao descompressor, ou seja, mais compressão.

Analisando as colunas que contém as PSNRs, pode-se verificar a vantagem obtida pelo método Tritree com reconstrução por interpolação e diferença entre o maior e o menor pixel, sobre as outras três combinações e visto que o método Tritree ainda se encontra em desenvolvimento pode-se esperar por ganhos ainda maiores em relação ao método Quadtree, que na concorrência com o método Tritree sempre perdeu em termos de PSNR, MSE, vértices e partições com mesma imagem e limiar de divisão.

9. Conclusão

Os resultados parciais confirmam a viabilidade do algoritmo TT para compressão de imagens coloridas, por produzir, simultaneamente, PSNR superior e número de vértices e partições inferior aos do algoritmo QT, um dos mais populares esquemas de codificação baseada em segmentação aplicado em compressão de imagens.

A utilização da decomposição TT nos mesmos métodos híbridos de compressão de imagens que fazem uso da QT geraria resultados superiores. As aplicações que fazem uso de decomposição ou segmentação de imagens, também poderiam utilizar a TT, como foi observado em imagens em níveis de cinza. Isto posta, a gama de possíveis aplicações da TT é consideravelmente ampliada.

10. Referências

- [1] Silva, V.C. e Carvalho, J.M. Image Compression by Tritree Decomposition. *Scientia - Unisinos*, 13(2):133–144, Julho/Dezembro 2002.
- [2] Bárbara, R.M.B e Alcaim, A. Compressão de Imagens Baseada em Decomposição Quadtree com Limiares Variáveis, Alocação de bits Adaptativa e DCT. SBT volume 38(3), March 1999.
- [3] Sullivan, G.J. e Baker, R.L. Efficient Quadtree Coding of Images and Video. *IEEE Transaction on Image Processing*, 3(3), May 1994.
- [4] Wille, S.O. Structured Tri-Tree Search Method for Generation of Optimal Unstructured Finite Element Grids in Two and Three Dimensions. *International Journal for Numerical Methods in Fluids*, 14:861–881, 1992.
- [5] Silva, V.C. Decomposição Triangular de Imagens: uma Aplicação em Compressão. Tese de doutorado,
- [6] Mayer, J. *Blending Models for Image Enhancement and Coding*. Tese de doutorado, Universidade da Califórnia, Santa Cruz, Dezembro 1999.
- [7] Cunha, T.O. e Silva, V.C. Decomposição Triangular de Imagens Coloridas: Uma Aplicação em Compressão. IV Semainainfo, Faculdade Metodista Granbery, Juiz de fora, MG, maio de 2007.
- [8] Cunha, T. O. ; Silva, V. C. . Compressão de Imagens Coloridas Utilizando Estruturas Tritree. In: VIII ERBASE - Escola Regional de Computação Bahia - Alagoas - Sergipe, 2008, Salvador/BA. Anais do VIII ERBASE - Escola Regional de Computação dos Estados da Bahia, Alagoas e Sergipe, 2008.

Redução de ghost em imagens de MR adquiridas pelo método EPI

Bruno C. Marini

(financiado pela FAPESP - Processo FAPESP No 2007/00930-7)

Graduação, Instituto de Computação - UNICAMP

nszeek@gmail.com

Marcelo V. W. Zibetti

Pós-doc, Instituto de Matemática, Estatística e Computação Científica - UNICAMP

marcelo.zibetti@terra.com.br

Gabriela Castellano (Orientadora)

Docente, Instituto de Física Gleb Wataghin - UNICAMP

gabriela@ifi.unicamp.br

Resumo

Este trabalho visou o estudo do artefato de Nyquist-Ghost (também conhecido como N/2-Ghost), característico de imagens de ressonância magnética (MR) adquiridas através do método de echo planar imaging (EPI), assim como a implementação de um método para sua correção. Para isso foram estudados vários métodos da literatura para a correção desse artefato, e foi feita uma implementação combinando alguns deles. O método implementado reduziu notavelmente o artefato nas imagens testadas.

hecido como Nyquist-Ghost. Isso se deve a que na aquisição das linhas do espaço-k (dados complexos), geralmente ocorre um erro de fase. Nas aquisições convencionais, como todas as linhas são adquiridas na mesma direção (p.ex., da esquerda para a direita), este erro de fase se torna comum a todo o espaço e aparece como um fator de escala global, que não afeta a reconstrução. Porém na aquisição EPI, linhas ímpares (adquiridas p.ex., da esquerda para a direita) terão fases diferentes das linhas pares (adquiridas da direita para a esquerda). Essa diferença de fase entre as linhas pares e ímpares dos dados causa o artefato tipo Ghost.

Em MR, o sinal medido $S(p, q)$ pode ser escrito como:

$$S(p, q) = \sum_{l,m} I(l, m) e^{-\frac{i2\pi lp}{N_x}} e^{-\frac{i2\pi mq}{N_y}}, \quad (1)$$

onde $k_x = p\Delta k_x$ e $k_y = q\Delta k_y$ são as coordenadas do espaço-k; $x = l\Delta x$ e $y = m\Delta y$ são as coordenadas no espaço da imagem; e $\Delta k_x = 1/(N_x\Delta x)$ e $\Delta k_y = 1/(N_y\Delta y)$. $I(l, m)$, que representa a imagem, é a magnetização da amostra, que depende da densidade de prótons e dos tempos de relaxação do tecido imageado [2]. Como mostra a Equação 1, esta imagem pode ser obtida do sinal medido fazendo a transformada de Fourier (FT) inversa deste sinal [1].

Considerando um erro de fase φ que se alterna entre as linhas pares e ímpares, podemos

1. Introdução

O método de EPI de aquisição de dados de MRI é muito mais rápido que os métodos convencionais de MR. Isto é útil em casos em que se necessita de alta resolução temporal, que é o caso dos experimentos de ressonância magnética funcional (fMRI), para estudo do funcionamento do cérebro.

O método de EPI [7] difere dos convencionais por preencher todo o espaço-k de aquisição dos dados com uma única excitação dos spins da amostra. Para apressar ainda mais a aquisição, o espaço-k é varrido em zig-zag (Figura 2), ao invés de seguir a varredura raster convencional (Figura 1). Essa varredura em zig-zag gera o artefato con-

re-escrever a Equação 1 como [9]:

$$S'(p, q) = \begin{cases} \sum_{l,m} I(l, m) e^{-\frac{i2\pi lp}{N_x} - \frac{i2\pi mq}{N_y} - i\varphi}, & q \text{ par} \\ \sum_{l,m} I(l, m) e^{-\frac{i2\pi lp}{N_x} - \frac{i2\pi mq}{N_y} + i\varphi}, & q \text{ ímpar} \end{cases} \quad (2)$$

Utilizando as propriedades de ortonormalidade da transformada de Fourier, é possível mostrar que a imagem I' reconstruída a partir de S' está relacionada à imagem I ideal (sem artefato) por [9]:

$$I'(l, m) = I(l, m) \cos \varphi + iI(l, m - \frac{N_y}{2}) \sin \varphi, \quad (3)$$

onde N_y é o número de pixels da matriz de imagem na direção de codificação de fase. Nesta equação, o primeiro termo representa a imagem “verdadeira”, com intensidade reduzida por um fator $|\cos \varphi|$; e o segundo termo corresponde ao ghost de Nyquist, cuja intensidade é dada por $I |\sin \varphi|$ e que está deslocada por $N_y/2$, ie, pela metade do tamanho da matriz de imagem na direção y . Exemplos deste artefato, junto com as respectivas imagens corrigidas (com o artefato reduzido), podem ser vistos nas Figuras 3, 4 e 5.

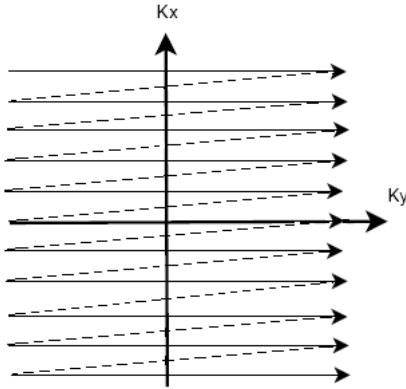


Figura 1. Aquisição regular

2. Métodos

A linguagem escolhida para a implementação do programa para a redução do artefato foi C pois esta é, em geral, mais rápida que as demais [3].

O método inicial escolhido para a redução do ghost foi o de Buonocore [8]. Neste método, ao invés de usar os dados do espaço-k para reconstruir a imagem via FT, duas imagens são reconstruídas, uma utilizando somente as linhas ímpares do espaço-k (e preenchendo as linhas pares com zeros antes de realizar a FT), e outra a partir das

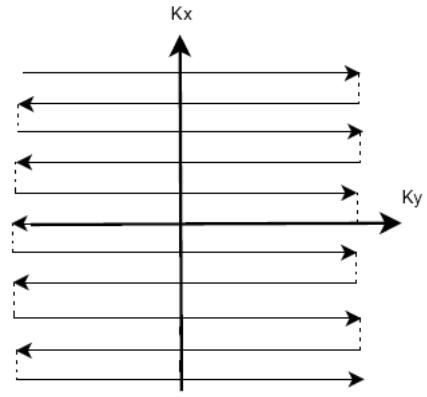


Figura 2. Aquisição EPI no espaço-k

linhas pares do espaço-k (preenchendo as ímpares com zeros). Desse modo, obtém-se duas imagens com ghost, mas um dos ghosts é positivo e o outro é negativo [8]. Se não houvesse erros de fase nas linhas pares e ímpares, então na região do objeto (região da imagem onde não há ghost) a diferença de fase entre as imagens par e ímpar seria zero. No caso em que não há sobreposição do ghost com a imagem, bastaria adicionar essas imagens ímpar e par e teríamos uma imagem corrigida. No caso de haver erros de fase, este pode ser detectado nas regiões de equivalência entre as imagens como uma diferença de fase. Pode-se, então, aplicar a correção em uma das imagens antes de realizar a soma entre elas, adicionando a nova fase em cada ponto (x, y) .

Quando o ghost se sobrepõe ao objeto de interesse, que é o caso mais comumente encontrado, ao invés de calcular uma diferença de fase para cada ponto da imagem, calcula-se uma diferença de fase média $\varphi(x)$ para cada linha x da imagem, onde a média é feita somente sobre as linhas que se encontram numa região (definida pelo usuário) que contém apenas o objeto (região sem ghost). A correção de fase então é feita no meio do caminho entre o espaço-k e o espaço da imagem, ie, aplica-se a FT apenas na direção de leitura [2], e faz-se a correção nessa “imagem” semi-transformada, multiplicando-se as linhas pares por $e^{-i\varphi(x)}$ e as linhas ímpares por $e^{i\varphi(x)}$.

No entanto, vemos que este método requer que o usuário selecione uma região de interesse na imagem. Como queríamos reduzir ao máximo a interação com o usuário, estudamos os métodos encontrados em [4], [5] e [6], e implementamos uma combinação destes com o método de [8].

Na implementação feita, calcula-se primeiro uma região de interesse automaticamente, onde

deveria estar a imagem “verdadeira”. Para isso, reconstrói-se a imagem normalmente aplicando a FT nos dados colhidos do espaço-k, gerando uma imagem F_{org} (com ghost). Como as imagens em questão são todas imagens de crânio (já que são para estudos do cérebro), a região de interesse (onde fica o crânio) é aproximadamente circular. Esta então é achada calculando-se o centro de massa da imagem F_{org} , e selecionado um círculo de diâmetro apenas menor que a menor dimensão da imagem, com centro no centro de massa da imagem (C_x, C_y) (usamos diâmetro $D = \min(W/1.25, H/1.25)$, onde W e H são as dimensões da matriz de imagem). Uma vez que temos essa região de interesse podemos aplicar o método de correção. Para isso, reconstroem-se as imagens par e ímpar como no método de Buonocore, F_{even} e F_{odd} . Estas imagens (complexas) são multiplicadas por uma fase composta de duas componentes, uma de zero ordem e uma de primeira ordem, e somadas para obter a imagem final:

$$F = F_{even}^{ft} e^{2\pi i(n_1 + n_2 x)} + F_{odd}^{ft} e^{-2\pi i(n_1 + n_2 x)}. \quad (4)$$

Para reduzir o ghost, queremos que os valores de F fora do círculo (fora da região da imagem “verdadeira”) sejam nulos. Portanto o método de correção é basicamente uma busca global pelas variáveis n_1 e n_2 que façam com que o valor de E seja mínimo, onde E é calculado como:

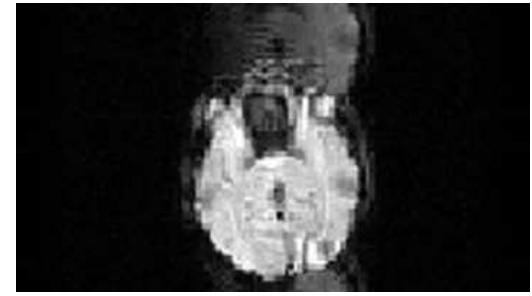
$$E = \sum_{x \in H} \sum_{y \in W} \sqrt{F_{real}(x, y)^2 + F_{img}(x, y)^2}. \quad (5)$$

3. Resultados

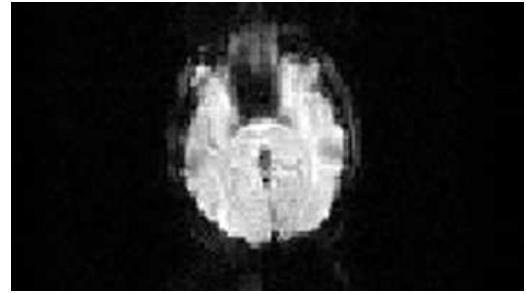
As Figuras 3, 4 e 5 mostram algumas das imagens testes, com ghost, e as respectivas imagens com o ghost reduzido, utilizando o método implementado.

4. Discussão e conclusões

Este trabalho teve por objetivo o estudo do artefato de Nyquist-Ghost em imagens de MR adquiridas pelo método de EPI, e a implementação de um método para a correção deste artefato. O método implementado, na linguagem C, foi uma combinação de diversos métodos existentes na literatura para a redução desse artefato, devido a que buscou-se um método que fosse simples e que tivesse a mínima interação possível com o usuário. Os resultados alcançados foram satisfatórios, percebe-se uma clara redução do artefato nas imagens testadas.

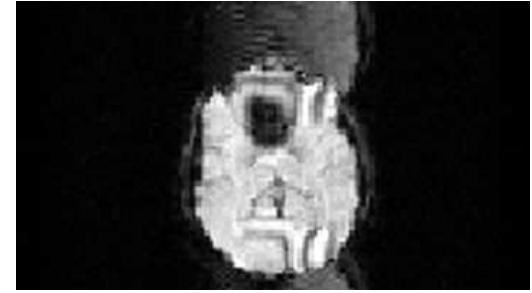


(a) Imagem original.

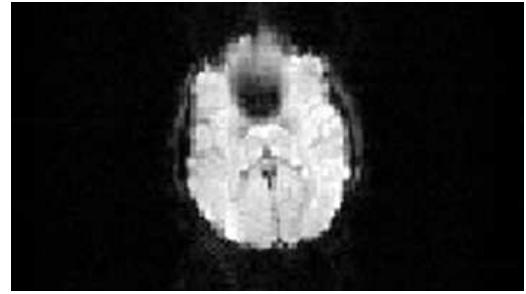


(b) Imagem corrigida.

Figura 3. Exemplo de imagem corrigida 1.



(a) Imagem original.

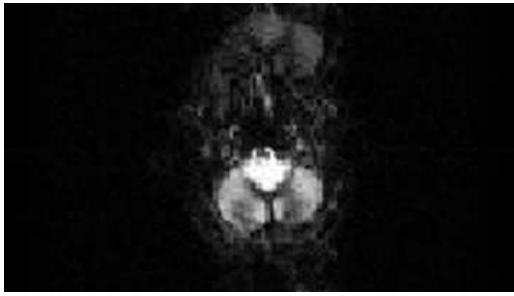


(b) Imagem corrigida.

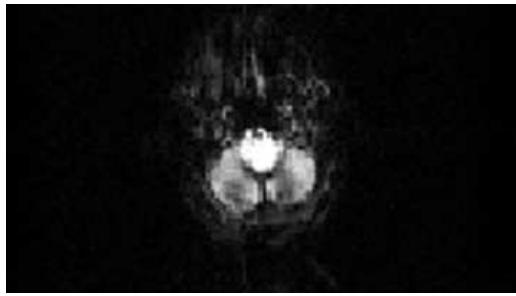
Figura 4. Exemplo de imagem corrigida 2.

Referências

[1] E. O. Brigham. The Fast Fourier Transform and its Applications. Prentice-Hall, Englewood Cliffs, New Jersey, 1988.



(a) Imagem original.



(b) Imagem corrigida.

Figura 5. Exemplo de imagem corrigida 3.

- [2] S. A. Huettel, A. W. Song, G. McCarthy. Functional Magnetic Resonance Imaging. Sinauer Associates, Sunderland, Massachusetts, USA, 2004.
- [3] W. H. Press et al. Numerical Recipes in C: The Art of Scientific Computing, 2^a ed. Cambridge University Press, 1992.
- [4] K. J. Lee. Image-Based EPI Ghost Correction Using an Algorithm Based on Projection Onto Convex Sets (POCS), 2002.
- [5] K. J. Lee. Image-based ghost reduction in interleaved EPI by method of generalised projections (MGP).
- [6] K. J. Lee. Image-based EPI ghost artifact reduction using iterative phase correction.
- [7] Robert L. De La Paz. Echo-planar Imaging.
- [8] Michael H. Buonocore, Lisheng Gao. Ghost Artifact Reduction for Echo Planar Imaging Using Image Phase Correction.
- [9] Bernstein MA, King KF, Zhou XJ. Handbook of MRI Pulse Sequences. Elsevier Academic Press, 2004.

Segmentação de Dentes Utilizando a Transformada Imagem-Floresta em Sistemas Biométricos

Elizabeth B. Barboza¹ (Bolsista FAPESP), Demiam O. Fonseca², Aparecido N. Marana³ (Orientador)

UNESP - Departamento de Computação – Faculdade de Ciências – Bauru
Av. Luiz Edmundo Carrijo Coube 14-01, Bauru-SP, 17033-360

{¹bethinha, ²demiam, ³nilceu}@fc.unesp.br

Abstract. *The positive human identification by dental arch radiographs is one of the most trustful biometric techniques. However, automatic dental segmentation from radiograph images is not an easy task and demands human interaction. A promising method to dental segmentation in computer biometrics systems, with minimum human intervention, is the Image Foresting Transform (IFT). The purpose of this work is to implement and to evaluate algorithms based on IFT for robust dental radiograph images segmentation.*

Resumo. *A identificação positiva de indivíduos utilizando radiografias da arcada dentária é uma das técnicas biométricas mais confiáveis. Porém, a segmentação automática dos dentes a partir de imagens de radiografias não é uma tarefa simples e exige interação humana. Um método promissor para segmentação de dentes em sistemas biométricos computacionais, com mínima intervenção humana, é a Transformada Imagem-Floresta (TIF). O objetivo desse trabalho é implementar e avaliar algoritmos baseados na TIF para segmentação robusta de dentes a partir de imagens de radiografias.*

1. Introdução

A identificação de indivíduos por meio de registros dentários é aceita pelas autoridades legais, pois é um método que fornece resultados precisos e confiáveis para identificação positiva, uma vez que as características extraídas são consideradas únicas e não são destruídas após a morte [1].

Nos sistemas biométricos computacionais baseados no reconhecimento de dentes a comparação das imagens ante-mortem (AM) e post-mortem (PM) é feita, em geral, utilizando-se descriptores da forma dos dentes. Portanto, é preciso segmentar os dentes nas radiografias e extrair seus contornos. A segmentação precisa dos dentes é uma das partes mais críticas dos sistemas biomé-

tricos baseados nas radiografias das arcadas dentárias [1].

O objetivo deste trabalho é implementar e avaliar algoritmos para segmentação de imagens de radiografias de arcadas dentárias, baseadas na Transformada Imagem-Floresta (TIF) [2], visando à detecção precisa dos contornos dos dentes, e de suas restaurações, com mínima intervenção humana. Os algoritmos de segmentação de imagens baseados na TIF computam, de forma robusta e eficiente, uma floresta de custo mínimo em um grafo derivado da imagem, através do uso de marcadores, por onde se inicia o crescimento da região [2].

2. Transformada Imagem-Floresta

A Transformada Imagem-Floresta (TIF) é uma ferramenta geral para modelar, implementar e avaliar operadores de processamento de imagens 2D/3D baseados em conexidade [2].

Uma imagem 2D/3D pode ser vista como um grafo onde os nós são os pixels e as arestas são definidas por uma relação de adjacência A entre os nós. Um caminho p nesse grafo é uma seqüência de nós distintos e adjacentes, e possui um valor de custo associado $c(p)$, cuja função de custo de caminho c depende de propriedades locais da imagem, tais como cor, gradiente e posição do nó ao longo do caminho. Um caminho p é ótimo se $c(p) \leq c(t)$ para qualquer outro caminho t com o mesmo destino de p . A busca por caminhos é realizada a partir de um conjunto dado S de nós sementes, determinadas pelo usuário. A TIF produz três atributos para cada pixel: seu predecessor no caminho ótimo, o custo desse caminho e o nó raiz correspondente (ou algum rótulo associado a ele) [2].

2.1. Abordagem Genérica

O algoritmo TIF genérico, ilustrado na Figura 1, visa encontrar um caminho ótimo, ou seja, um caminho com um custo mínimo no grafo derivado da imagem, que se origina no conjunto de sementes dado [2].

Quando existem várias sementes em S , estas são propagadas ao mesmo tempo, estabelecendo um processo competitivo. Cada semente irá definir uma zona de influência composta por pixels conexos a ela por caminhos mais baratos do que os fornecidos por qualquer outra semente em S [2].

ENTRADA: Uma Imagem com pixels de intensidade I , relação de adjacência A e um conjunto de nós sementes S .

SAÍDA: Mapa de Custo C , mapa de Predecessores P e mapa de Raízes R .

Para Cada nó p do grafo derivado da imagem Faça

$C(p) \leftarrow +\infty$, $R(p) \leftarrow p$, $P(p) \leftarrow \text{nil}$.

Para Cada nó $p \in S$ Faça

$C(p) \leftarrow 0$

 Insira p em Q

Enquanto Q não estiver vazia, Faça

 Remova p de Q tal que $C(p)$ seja mínimo

 Para Cada nó $q \in A(p)$ tal que $C(q) > C(p)$ Faça

$cst \leftarrow \max(C(p), |I(q) - I(p)|)$

 Se $cst < C(q)$ Então

 Se $C(q) \neq +\infty$ Então Remova q de Q

$C(q) \leftarrow cst$, $R(q) \leftarrow R(p)$, $P(q) \leftarrow p$

 Insira q em Q

Retorne $\{C, P, R\}$.

Figura 1. Algoritmo TIF Genérico. Fonte [2].

2.2. Segmentação por Competição entre Sementes K-Conexas

Um conjunto de pixels é dito componente k-conexo com relação a uma dada semente, quando este é alcançado por caminhos ótimos com custos inferiores ou iguais a um valor k . Um componente k-conexo com relação a um pixel semente pode ser redefinido como um conjunto de pixels com forças de conexidade em relação a sementes maiores ou iguais a um limiar derivado de k [2].

A idéia deste método é limitar o tamanho máximo das frentes de mesmo custo, fazendo com que seus crescimentos parem quando ultrapassam um dado limiar. Isto faz com que o problema de achar um k para cada semente se reduza a um único limiar de área aplicado às frentes de custo. O método restringe a competição entre sementes em regiões com pixels que são k-conexos a alguma semente s , o que o torna mais efetivo. Pixels não alcançados por nenhuma semente são considerados como pertencentes ao fundo e sementes externas são necessárias somente quando há vazamentos [2].

3. Resultados Experimentais

Na extração do contorno de dentes utilizando a abordagem genérica do algoritmo TIF (Figura 2(a)) nota-se a concorrência entre sementes, o que gera bordas nos limites de abrangência de cada pixel semente. Para a mesma imagem, utilizando o algoritmo de segmentação por Competição entre Sementes K-Conexas, os resultados são melhores comparados com o método anterior. Como a competição entre as sementes depende de um valor k distinto, eliminou-se a concorrência entre sementes pertencentes a uma mesma região, já que há diferenciação entre sementes pertencentes ao objeto e ao fundo (Figura 2(b)).

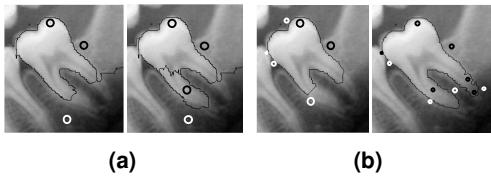


Figura 2. Segmentação de dente: (a) TIF Genérico; (b) Segmentação por Competição entre Sementes K-Conexas. Os círculos representam as sementes.

Para o caso da extração do contorno de restaurações, foram aplicadas as mesmas técnicas para segmentação dos dentes. A utilização do algoritmo de segmentação por Competição entre Sementes K-Conexas se mostrou mais efetiva do que a abordagem genérica (Figuras 3(a) e 3(c)). Para realçar bordas de interesse foi utilizado um pré-processamento baseado no operador Sobel. Observa-se que com o pré-processamento houve uma melhoria nos resultados das segmentações, pois os contornos foram suavizados (Figuras 3(b) e 3(d)).

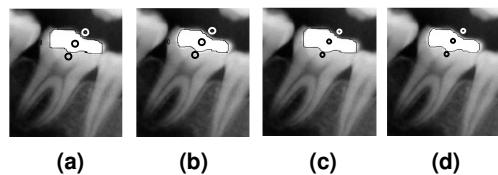


Figura 3. Segmentação de restauração: (a) TIF Genérico; (b) TIF Genérico e Sobel; (c) Competição entre Sementes K-Conexas; (d) Competição entre Sementes K-Conexas e Sobel. Os círculos representam as sementes.

Para avaliar o desempenho dos algoritmos de segmentação, foi realizada a detecção manual dos contornos dos dentes (*ground truth*), utilizando um software de manipulação de imagens. Os contornos gerados pelos algoritmos foram comparados visualmente com os manuais. As Figuras 4(a) e 4(b) mostram exemplos de segmentações obtidas pelo método genérico da TIF (esquerda), ao lado de suas respectivas segmentações manuais (direita). A Figura 4(c) apresenta o contorno da restauração gerado pelo algoritmo de Segmentação por Competição entre Sementes K-Conexas. É possível observar que houve um bom desempenho nas segmentações obtidas pelos métodos TIF, comparando-se com as segmentações manuais.

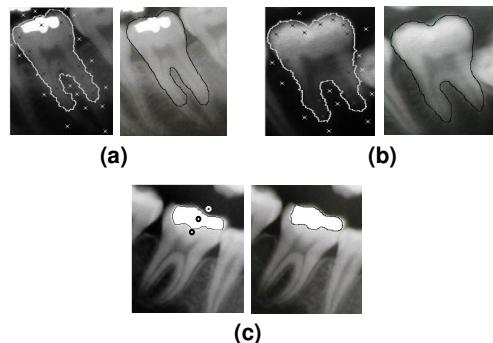


Figura 4. Análise qualitativa: (a) e (b) Segmentação de dente com o algoritmo TIF Genérico (esquerda); (c) Segmentação de restauração com o algoritmo de segmentação por Competição entre Sementes K-Conexas. À direita de cada imagem estão as segmentações manuais. Os pontos e os círculos representam as sementes.

Para a análise quantitativa foi realizado um experimento com uma amostra de 9 indivíduos, sendo duas radiografias para cada, obtidas em intervalos de um a quatro anos entre elas. A radiografia mais antiga foi considerada como sendo a AM e a mais nova a PM. Cada teste comparativo foi realizado entre a segmentação do terceiro molar inferior esquerdo de cada radiografia e os mesmos molares dos outros indivíduos, além de sua radiografia genuína, fazendo uso da abordagem genérica da TIF. Foram utilizados 100 pontos de comparação entre as formas, com o algoritmo *Shape Context* [3] para descrição e comparação das formas. A Figura 5 mostra os gráficos obtidos nesta avaliação. As distâncias foram normalizadas no intervalo de 0 a 20. A taxa de erro obtida (Equal Error Rate - EER) para este teste foi de aproximadamente 26%. Esse resultado pode ser considerado razoável, principalmente porque essa taxa de erro foi obtida utilizando-se apenas um dos dentes da arcada dentária.

4. Conclusão

Os algoritmos implementados para segmentação de dentes baseados na Transformada Imagem Floresta (TIF) mostraram-se satisfatórios, principalmente para a segmentação das restaurações. A segmentação por Competição entre Sementes K-Conexas apresentou melhores resultados do que a abordagem genérica. Os resultados preliminares obtidos indicam que algoritmos baseados em TIF são uma alternativa promissora para sistemas biométricos computacionais de reconhecimento de dentes, uma vez que permitem a segmentação eficaz dos dentes e suas restaurações com mínima intervenção do usuário.

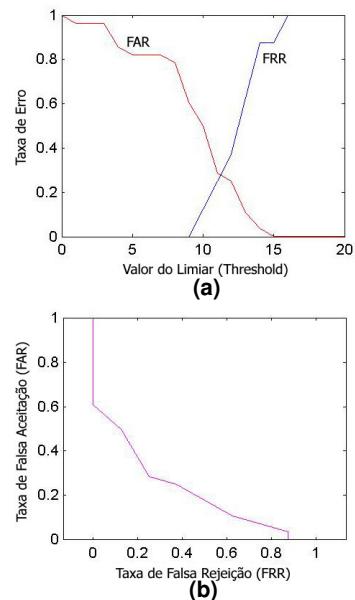


Figura 5. Análise quantitativa: (a) Curvas da taxa de falsa aceitação (FAR) versus taxa de falsa rejeição (FRR). Observa-se o valor da taxa de erro (EER) de aproximadamente 26%; (b) Curva ROC.

Agradecimentos

Os autores agradecem à Fapesp (processo no 2007/02662-0) pelo auxílio financeiro.

Referências

- [1] Jain, A. K.; Chen, H.; and Minut, S. (2003). “Dental Biometrics: Human Identification Using Dental Radiographs”, Proceedings of 4th Int. Conf. AVBPA, Guildford, UK, pages 429-437, June.
- [2] Miranda, P. (2006). “Segmentação de Imagens pela Transformada Imagem-Floresta”, Dissertação (Mestrado em Ciência da Computação) – UNICAMP. Orientador: Alexandre Xavier Falcão.
- [3] Belongie, S., Malik, J., Puzicha, J. (2002). “Shape Matching and Object Recognition Using Shape Contexts”, IEEE Trans. of PAMI, vol. 24, n. 4, pp. 509-522.

Geração da Curva Interseção de Superfícies Trianguladas

André Luiz Firmino Alves

Rosana Marques da Silva (Orientadora)

Universidade Federal de Campina Grande – UFCG

Departamento de Matemática e Estatística - DME

andre.alfa@gmail.com, rosana@dme.ufcg.edu.br

Resumo

O presente trabalho apresenta uma implementação do algoritmo para a obtenção da curva de interseção de duas superfícies trianguladas, apresentado por Lo e Wang [3]. O algoritmo determina a curva de interseção entre duas malhas trianguladas criando novos vértices em cada uma das superfícies, gerando uma poligonal pertencente a ambas as superfícies, as quais devem ser reconstruídas localmente a partir desses novos vértices, recortando as faces excedentes. O problema de interseção entre duas superfícies trianguladas é tratado como um problema de interseção entre dois triângulos e dividido nas seguintes etapas: Determinação do ponto de interseção entre o plano que contém uma das faces e uma aresta da outra face e a verificação da pertinência do ponto de interseção determinado a ambos os triângulos. A eficiência do algoritmo é melhorada pelo uso de caixas limitantes.

1. Introdução

A operação de interseção entre superfícies modeladas por malhas poligonais, para a representação de estruturas arbitrárias e complexas, requer algoritmos de alta performance, visto que as interseções, geralmente, ocorrem em pequenas regiões das superfícies. Assim, é desejável a definição de um processo preciso e eficiente para determinar as curvas de interseções.

Através da curva de interseção de duas superfícies poliédricas pode-se construir uma única malha a partir da junção das malhas que modelam cada uma das superfícies envolvidas, onde a malha resultante deve ser formada pelas malhas iniciais modificadas localmente nas regiões onde ocorrem as interseções.

As operações de interseção e união permitem a construção de modelos geométricos de superfícies complexas, a partir de modelos de superfícies mais simples, construídos individualmente [2,3].

Este trabalho apresenta uma implementação do algoritmo de interseção de superfícies, representadas por malhas trianguladas, apresentado por Lo e Wang [3]. O algoritmo determina a curva de interseção entre duas malhas trianguladas criando novos vértices, em cada uma das superfícies, gerando uma poligonal pertencente a ambas as superfícies, as quais devem ser reconstruídas, localmente, a partir desses novos vértices, recortando as faces excedentes.

O cálculo da curva de interseção em superfícies trianguladas, apresentado neste trabalho, foi utilizado no desenvolvimento de um protótipo de modelador de cenários geológicos tridimensionais de reservatórios petrolíferos [1]. Os volumes conexos que formam um reservatório têm formas arbitrárias e complexas, e podem ser obtidos seguindo o método da superposição das unidades geológicas que compõem o reservatório, através das operações de interseção entre essas unidades modeladas individualmente como superfícies trianguladas [5].

2. Superfícies Trianguladas

A utilização de faces triangulares na poligonização de superfícies apresenta algumas vantagens, entre elas: Planaridade, que facilita cálculos entre superfícies, tais como as interseções, que podem ser vistas como interseções entre retas (arestas) e planos (triângulos); e Sistema de coordenadas, visto que um triângulo tem associado a ele, naturalmente, um sistema de coordenadas lineares (coordenadas baricêtricas) [4].

3. Interseção entre Faces Triangulares

A interseção entre duas superfícies trianguladas pode ser tratada como um problema de interseção entre triângulos (Figura 1), podendo ser dividido nas seguintes etapas:

- 1- Determinação do ponto de interseção entre o plano que contém uma das faces e uma aresta da outra face;
- 2- Verificação se o ponto de interseção determinado pertence a ambos os triângulos.

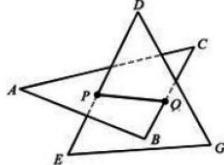


Figura 1. Interseção entre dois triângulos [3]

3.1. Interseção entre os triângulos ABC e DEG.

O problema de interseção entre uma face triangular ABC com outra face triangular DEG , como já mencionado, pode ser tratado como um problema de interseção entre planos e retas. Mais precisamente deve-se verificar a ocorrência de interseção das arestas DE , EG e GD com a face triangular ABC (Figura 1). Para verificar se a aresta DE intercepta o triângulo ABC as seguintes condições devem ser satisfeitas [3]:

1. Os pontos D e E pertencem a lados opostos do plano que contém o triângulo ABC , ou seja,

$$(\overrightarrow{AD} \cdot \overrightarrow{N})(\overrightarrow{AE} \cdot \overrightarrow{N}) < 0, \quad (1)$$

onde $\overrightarrow{N} = \overrightarrow{AB} \times \overrightarrow{AC}$ é o vetor normal ao plano que contém a face ABC .

2. O ponto P de interseção está contido no triângulo ABC .

Dado que a condição anterior é satisfeita, tem se que a aresta DE corta o plano que contém o triângulo ABC . Então basta verificar se o ponto P de interseção da aresta DE com o plano está contido no triângulo ABC . Observe que além de P pertencer a aresta (reta) DE , P também pertence ao plano que contém o triângulo ABC , logo P satisfaz o sistema de equações:

$$\begin{cases} P = tE + (1-t)D \\ (A _ P) \cdot N = 0 \end{cases} \quad (2)$$

Assim o parâmetro t é dado por

$$t = \frac{\overrightarrow{AD} \cdot \overrightarrow{N}}{\overrightarrow{AD} \cdot \overrightarrow{N} - \overrightarrow{AE} \cdot \overrightarrow{N}}. \quad (3)$$

3.2. Verificando se o ponto P pertence ao triângulo ABC

O ponto P estará dentro do triângulo ABC se

$$(\overrightarrow{AB} \times \overrightarrow{AP}) \cdot \overrightarrow{N} > 0, (\overrightarrow{BC} \times \overrightarrow{BP}) \cdot \overrightarrow{N} > 0, (\overrightarrow{CA} \times \overrightarrow{CP}) \cdot \overrightarrow{N} > 0. \quad (4)$$

De forma análoga, pode-se verificar se as arestas EG e GD interceptam o triângulo ABC (Figura 1), para o cálculo do segundo ponto da interseção. Se as outras arestas do triângulo DEG não interceptam o triângulo ABC , então o segundo ponto de interseção pode ser determinado calculando a interseção das arestas do triângulo ABC (AB , BC e CA) com o triângulo DEG . Seja Q o segundo ponto da interseção, logo o segmento PQ compõe a curva de interseção entre as duas faces triangulares ABC e DEG .

A Figura 2 mostra os resultados da implementação realizada para os cálculos da verificação de interseção entre faces triangulares (Figura 2.a) a fim de determinar a curva de interseção entre os dois triângulos: determinação dos pontos de interseção (Figura 2.b) e do segmento da curva de interseção (Figura 2.c).

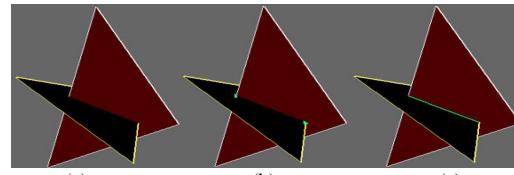


Figura 2. Interseção entre dois triângulos [1].

3.3. Casos previstos nas interseções dos triângulos

Na interseção entre dois triângulos, existem pelo menos quatro casos possíveis (Figura 3) [3]:

1. O ponto de interseção P ocorre dentro do triângulo ABC e pertence a uma das arestas do triângulo DEG (Figura 2 e 3.a);
2. Os pontos de interseção P e Q pertencem as arestas do triângulo ABC (Figura 3.b);
3. O ponto de interseção P coincide com um vértice do triângulo ABC (Figura 3.c);
4. A interseção não é mais um ponto, e sim um domínio planar, ou seja, o produto vetorial entre os vetores normais dos triângulos ABC e DEG é nulo, pois os triângulos estão no mesmo plano (Figura 3.d).

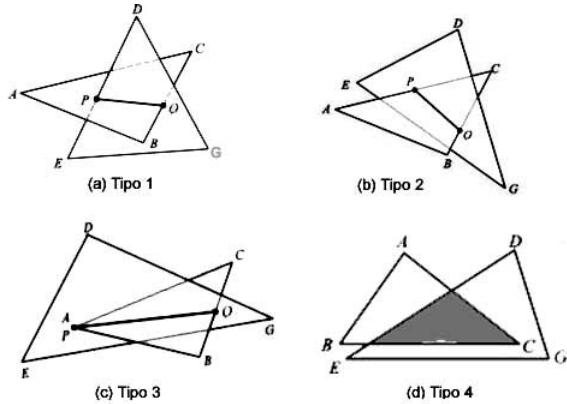


Figura 3. Possibilidades de interseção entre dois triângulos

4. Determinação da Curva de Interseção

A interseção entre duas superfícies normalmente ocorre apenas em algumas de suas regiões, necessitando-se, portanto, em termos de eficiência computacional, de um mecanismo para filtrar as faces onde não há possibilidade de ocorrência de interseção (Figura 4). Lo e Wang [3] utilizam a técnica da Caixa Limitante para reduzir o tempo de pesquisa das faces trianguladas, filtrando as faces onde não ocorrem interseções nas superfícies. A utilização de Caixas Limitantes reduz drasticamente o tempo de checagem das interseções entre as faces trianguladas, tornando o tempo do processo de interseção quase linear em relação ao número de elementos (faces) envolvidos.

O algoritmo utiliza uma Caixa Limitante, chamada neste trabalho de Caixa Limitante Global, para encapsular os objetos (malhas), ou parte dos objetos envolvidos no processo, e caixas limitantes para cada face triangulada de cada superfície, chamadas de *bounding box*. A Caixa Limitante Global trata-se de uma partição espacial (Figura 5.a) em células de dimensões menores, doravante chamadas de *voxels* (Figura 5.b). As dimensões dos *voxels* podem ser arbitrárias, como também ajustadas dinamicamente de acordo com o tamanho dos objetos. Lo e Wang [3] utilizam *voxels* de mesma dimensão e cada *voxel* poderá conter mais de um objeto (face triangular).

5. Implementação do algoritmo

Esta seção apresenta uma implementação do algoritmo apresentado por Lo e Wang [3], com algum detalhamento.

Dadas duas superfícies trianguladas S_1 e S_2 (Figura 5), segue os passos do algoritmo implementado:

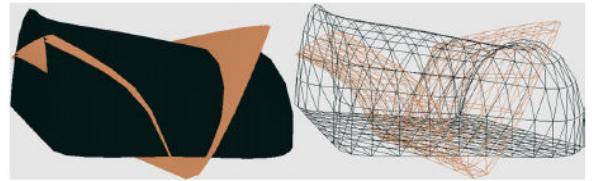


Figura 4. Interseção entre superfícies poliédricas

1. Definir a Caixa Limitante Global (CLG) (Figura 5.a), os *Bounding Box's* das faces triangulares (Figura 6) e determinar as partições da CLG, ou seja, os *voxels* V_i (Figura 5.b);

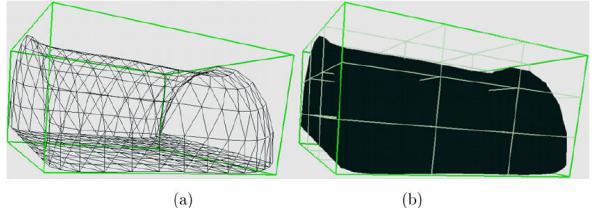


Figura 5. (a) CLG de uma superfície (b) Os Voxels.

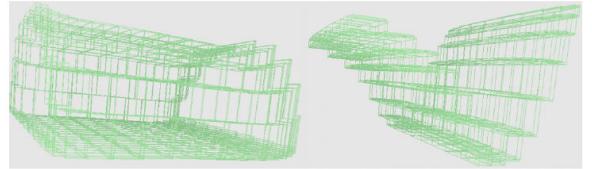


Figura 6. *Bounding box's* das faces trianguladas das superfícies S_1 e S_2 .

2. Identificar as interseções entre os *voxels* (Figura 5.b) e os *bounding box's* (Figura 6) das faces triangulares, ou seja, identificar as faces T_k em S_1 que interceptam os *voxels* V_i , e as faces F_j em S_2 que interceptam os *voxels* V_i (Figura 7.a):

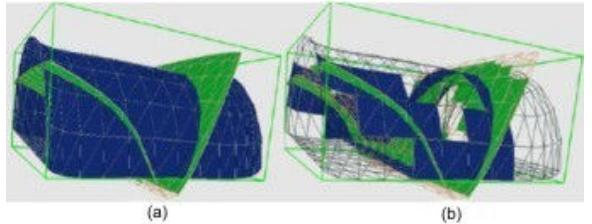


Figura 7. As faces candidatas a interseção.

3. Selecionar faces candidatas (Figura 7.b):
 - (a) Cada *voxel* V_i (Figura 5b) é examinado por vez. V_i é ignorado se o *bounding box* (Figura 6) da face T_k em S_1 ou o *bounding box* da face F_j em S_2 não está em seu interior.
 - (b) Se V_i contém o *bounding box* de uma face T_k em S_1 e o *bounding box* de uma face F_j em S_2 , e os *bounding box* dessas faces se interceptam,

então essas faces são consideradas candidatas a interseção (Figura 7.b e Figura 8).

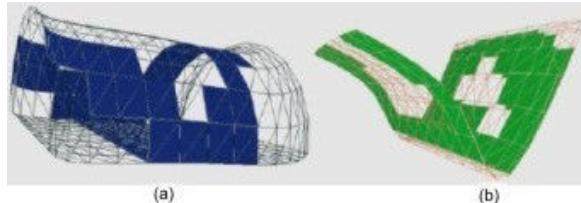


Figura 8. As faces candidatas a interseção em cada uma das superfícies.

4. Determinar a curva de interseção: Para todos os $voxels V_i$ que contém faces candidatas a interseção:
 - (a) Calcular o ponto de interseção entre as faces candidatas de ambas as superfícies
 - (b) Se o ponto de interseção entre as faces candidatas de ambas as superfícies pertencerem as faces, então registra os pontos nas respectivas faces.
 - (c) Criar a curva poligonal a partir dos pontos de interseção (Figura 9).

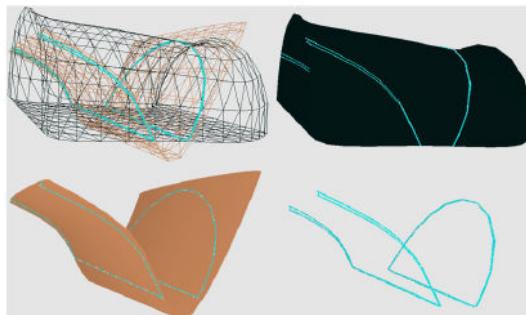


Figura 9. Obtenção da curva de interseção.

A implementação do algoritmo que determina a curva de interseção em malhas trianguladas foi realizada em linguagem C, na plataforma Linux, utilizando funções da biblioteca TDM (Módulo de Dados Topológicos), desenvolvida por Wu [6] para representar e gerenciar o modelo das superfícies, e funções da biblioteca Mesa/OpenGL para auxiliar a visualização do modelo gerado.

5. Considerações Finais

Este trabalho apresentou uma implementação do algoritmo, apresentado por Lo e Wang [3], para obtenção de uma poligonal que representa a curva interseção entre duas superfícies trianguladas. O algoritmo usa caixas limitantes, reduzindo, com isso, significativamente o tempo para identificar as regiões

em que há possibilidade de ocorrência de interseções. Geralmente as interseções entre superfícies ocorrem em pequenas regiões, portanto o procedimento que utiliza as caixas limitantes é bastante eficiente.

Um próximo passo para a obtenção de uma única malha através da operação de interseção entre duas superfícies seria uma reconstrução local na região que contém a curva de interseção das superfícies. Além do mais o algoritmo implementado deve ser adaptado aos casos de interseção de n superfícies ($n > 2$) e superfícies cuja curva interseção contém múltiplos laços (auto-interseções).

6. Agradecimentos

À ANP/PRH-25 pelo apoio financeiro e ao Departamento de Matemática e Estatística/UFCG pelo suporte para a realização deste trabalho.

7. Referências

- [1] ALVES, A. L. F. Geração de Cenários Tridimensionais de Reservatórios Petrolíferos Canalizados. Monografia de Graduação. UFCG/CCT/DME/PRH-25, 2007.
- [2] COELHO, L. C., GATTASS, M. Intersecting and trimming parametric meshes on finite element shells. International Journal for Numerical Methods in Engineering, vol.0(0), 1-10, 1999.
- [3] LO, S. H., WANG, W. X. A Fast Robust Algorithm for the Intersection of Triangulated Surfaces. Engineering with Computers, 2004.
- [4] MIRANDA, A. C. O., MARTHA, L. F.. Geração de malhas estruturadas a partir de um espaço paramétrico de triangulações não estruturadas. Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Rio de Janeiro, 2004.
- [5] SANTOS, J. A. ARAÚJO, C. E. S. e SILVA, R. M.. Geração de um Modelo Geométrico de Canais Encontrados em Reservatórios Petrolíferos. XVIII Brazilian Symposium on Computer Graphics and Image Processing, SIBGRAPI - WORKSHOP of Undergraduate Work, Natal, RN, 2005.
- [6] WU, S.-T. Non-manifold Data Models: Implementational Issues. In Proceedings of MICAD'92, pg 38-56, Paris, 1992.

Método Híbrido de Reconhecimento Ótico de Caracteres

William Ivanski, Luciano Silva, Olga R. P. Bellon
{wiva06, luciano, olga}@inf.ufpr.br

Grupo IMAGO de Pesquisa*† – UFPR, Cx.P. 19902 – 81531-990 – Curitiba-PR

Resumo

Com os recentes avanços nas áreas de Processamento de Imagens e Visão Computacional, reconhecer caracteres em imagens tem se tornado viável para uma variedade cada vez maior de aplicações, incluindo a integração entre OCR (Optical Character Recognition) e ferramentas de acessibilidade. Este trabalho apresenta um sistema de OCR híbrido, baseado em Análise Estrutural e Comparação de Modelos, e ainda conta com um método de segmentação de caracteres conexos. Para mostrar a efetividade da solução proposta, foi realizado um experimento comparativo utilizando outros dois sistemas de OCR para software livre.

1. Introdução

Reconhecimento Ótico de Caracteres, ou no inglês *Optical Character Recognition* (OCR) [2, 10, 13], é uma das mais importantes áreas de Visão Computacional. O OCR consiste em extraír texto de imagens e, através de um algoritmo de classificação, gerar um texto editável equivalente ao texto contido na imagem. É usado em *software* de *scanners* de mesa, que digitalizam um documento e reconhecem o texto nele contido, permitindo que o usuário altere o documento da forma que desejar [4]. Para isso, o sistema OCR deve ser capaz de processar imagens degradadas, com ruído, apresentando textos com fontes inclinadas ou distorcidas, em diferentes formatos e resoluções [4, 9].

Recentemente, OCR também tem sido usado como ferramenta de acessibilidade, integrado a outras tecnologias assistivas, tais como leitores e magnificadores de tela [5]. Para tal finalidade, um sistema OCR ideal deve conseguir reconhecer texto com fontes nos tamanhos de 10 a 12, mais usados em aplicações *Desktop*, e nos tamanhos de 12 a 14, mais usados na *Web* [7].

De acordo com o método de extração de características utilizado na fase de reconhecimento, os sistemas

OCR podem ser divididos em duas categorias: os sistemas baseados em Análise Estrutural e os sistemas baseados em Comparação de Modelos [11]. A Análise Estrutural extrai de cada caractere um conjunto de características simples, também chamadas de descritores, que formam a representação estrutural do caractere [11]. Como exemplos de sistemas baseados somente em Análise Estrutural, temos o *gocr*¹, o *ocrad*² e o *tesseract*³. A Comparação de Modelos utiliza apenas uma matriz, o chamado modelo, para representar a forma do caractere, de acordo com o posicionamento de seus pontos [11]. Como exemplos de sistemas que utilizam somente Comparação de Modelos, temos o *calera* [2] e o sistema descrito em [12, 13].

Os sistemas que baseiam-se em somente uma dessas categorias são chamados puros, e os sistemas que baseiam-se nas duas são chamados híbridos [11]. O problema de utilizar métodos puros de reconhecimento é que o número de amostras de treinamento da base de dados deve ser muito grande para que o sistema atinja a robustez. O *tesseract*, por exemplo, utiliza 60.160 amostras em sua fase de treinamento [10], e um algoritmo de reconhecimento detalhado em [1] utiliza 1.175.000 amostras. Ainda, se o método de reconhecimento for baseado somente em Análise Estrutural, o número de características deve ser muito grande. O *tesseract* extrai entre 50 e 100 características para cada caractere [10]. Outro problema da Análise Estrutural pura é a intolerância a variações estruturais no caractere, causadas devido à degradação ou à baixa qualidade da imagem [2].

Neste trabalho é apresentado um método híbrido que explora as propriedades da Análise Estrutural e da Comparação de Modelos. Da união destes dois métodos, o método proposto possui as vantagens de utilizar poucos descritores e menos amostras na fase de treinamento, como será apresentado nas seções seguintes. Para a implementação do sistema OCR proposto, denominado *imagocr*, foram utilizados algoritmos clássicos encontrados na literatura, porém agregando novas e importantes contribuições: (1) algoritmo de corte para segmentação

* www.imago.ufpr.br

† Os autores gostariam de agradecer ao CNPq e a FINEP pelo suporte financeiro.

1 <http://sourceforge.net/projects/jocr/>

2 <http://www.gnu.org/software/ocrad/>

3 <http://sourceforge.net/projects/tesseract-ocr/>

de caracteres conexos; (2) algoritmo de segmentação de palavras; (3) conjunto de descritores; (4) novo algoritmo de reconhecimento, que baseia-se no conjunto de descritores. O objetivo da implementação deste sistema é a sua aplicação em ferramentas de acessibilidade para pessoas com necessidades especiais [5].

2. Visão Geral do Sistema

O sistema OCR possui quatro etapas principais: (1) Aquisição da Imagem, (2) Pré-Processamento, (3) Segmentação e (4) Reconhecimento.

2.1. Aquisição da Imagem e Pré-Processamento

As imagens de entrada do sistema são imagens de ambientes Desktop e Web, capturadas com o auxílio da ferramenta *MouseLupa* [5]. A princípio, consideramos apenas imagens com fundo bem comportado, sem a presença de figuras e sem ruído ou degradação. Foi desenvolvida uma pequena biblioteca para processar essa imagens.

Uma vez carregada na memória, a imagem passa pela etapa de pré-processamento, onde é convertida para escala de cinza e em seguida é binarizada utilizando o algoritmo de Otsu [8], que encontra um limiar de binarização automaticamente. Não foi usado um algoritmo de binarização com limiar fixo porque diversas fontes representam caracteres pequenos utilizando fatores de cinza muito menores do que os utilizados em caracteres maiores, representados pela mesma fonte.

2.2. Segmentação

Através da projeção horizontal da imagem binária, são detectadas linhas de texto em um documento [3], isolando linha por linha. Para isolar os caracteres em cada linha, poderia ser utilizada projeção vertical [3] ou rotulação de componentes conexos [4]. A segunda facilita a extração de caracteres em fundos com uma certa variação, e generaliza o processo de reconhecimento. De agora em diante, cada caractere extraído de uma linha, através de rotulação, será chamado de objeto.

Quando a fonte é muito pequena, usualmente no texto se encontram caracteres que se tocam, *i.e.*, estão “grudados”, como apresentado na Fig. 1. Nesses casos, o objeto representa mais de um caractere, de forma que não se pode reconhecer tal objeto utilizando o processo de reconhecimento a nível de caractere. Na literatura são encontrados muitos métodos que tratam desse problema com muita eficiência quando aplicados a fontes de largura fixa. Entretanto, o problema ainda está em aberto quando são tratadas fontes de largura proporcional ou variável [3].

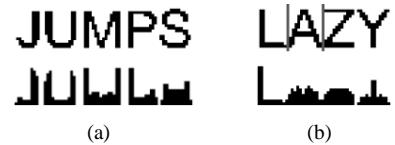


Figura 1. Exemplos de palavras e suas projeções verticais: (a) caracteres que não se tocam, e que são facilmente segmentados através de rotulação; (b) caracteres que se tocam, e que são segmentados com o auxílio de pontos de corte (linhas verticais delimitando o caractere “A”).

Para segmentar caracteres grudados, foi utilizado um método de corte que utiliza projeção vertical (veja Fig. 1), e é baseado em reconhecimento [3, 9, 12]. Tal método encontra um ou mais pontos de corte em um dado objeto analisando a projeção vertical desse objeto, e em seguida decide se os pontos de corte estão corretos ou não, através do reconhecimento das partes isoladamente. Esse processo, chamado janelas deslizantes, é feito com o auxílio de uma estrutura de dados (árvore), e cada parte reconhecida isoladamente é chamada de janela. O caminho com a maior probabilidade, da raíz da árvore até suas folhas, é escolhido como o correto [9, 12].

Como os pontos de corte sempre são os pontos mínimos na projeção vertical do objeto [3, 9], teríamos pontos de corte no meio de caracteres tais como O, o, e, B, e a, dentre outros. Por essa razão, foi acrescentada ao método de corte uma condição a mais na escolha de pontos de corte. Junto com a projeção vertical, também é obtida uma projeção vertical de transformações, que contabiliza, para cada coluna, a soma das transformações de 0 para 1. Então, pontos mínimos na projeção vertical do objeto não são considerados pontos de corte se, naquele ponto, o valor da projeção vertical de transformações for maior do que 1. Isso elimina os cortes desnecessários no meio de caracteres.

As distâncias entre os caracteres atual e anterior são usadas para segmentar palavras, o que significa saber onde inserir espaços em branco em uma seqüência de caracteres reconhecidos. A solução adotada reside no fato de que, dentro de um conjunto de distâncias entre caracteres, os espaçamentos entre palavras são sempre maiores do que espaçamentos entre caracteres dentro de uma mesma palavra. Isso pode ser feito usando o algoritmo de Otsu [8], que encontra um limiar divisor entre dois grupos de valores. Se a distância entre o caractere atual e o anterior for maior do que o limiar, então temos um espaçamento entre palavras; caso contrário, temos um espaçamento simples entre caracteres numa mesma palavra.

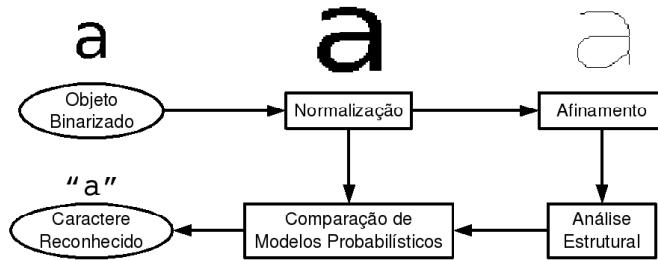


Figura 2. Diagrama do sistema proposto para o reconhecimento de caracteres.

2.3. Reconhecimento

Quanto menos características forem usadas na Análise Estrutural, menos eficiente é o método de reconhecimento, pois faltam elementos discriminantes para diferenciar caracteres parecidos. Além disso, quanto menos amostras forem utilizadas para treinar um método de Comparação de Modelos, menos robusto será o reconhecimento, pois os modelos da base não estarão aptos a se adaptar a determinadas variações na forma do caractere não previstas pela fase de treinamento. Em ambos os casos, o algoritmo de reconhecimento não saberá discernir corretamente determinados caracteres parecidos.

A estratégia adotada no sistema proposto explora os pontos fracos e fortes da Análise Estrutural e da Comparação de Modelos, através de um método híbrido, que combina os dois métodos. O reconhecimento foi dividido em duas fases: uma fase de Análise Estrutural, que utiliza menos descriptores do que um método puro, e outra fase de Comparação de Modelos Probabilísticos. Outro ponto importante do método proposto é usar um número muito menor de amostras na fase de treinamento: foram utilizadas 35 amostras de 86 caracteres, totalizando 3010 amostras.

O algoritmo de reconhecimento de caracteres é ilustrado na Fig. 2. Primeiro, a matriz binarizada do objeto é normalizada, o que significa que ela é redimensionada para o tamanho dos caracteres contidos na base de dados [4]. Em seguida, ela é afinada, utilizando o algoritmo de esqueletização descrito em [9]. A partir do esqueleto, são extraídas as seguintes características: (1) acentuação: diz se o caractere é acentuado ou não, a partir da análise do objeto imediatamente superior ao caractere em questão; (2) medianas: número de cortes nos eixos vertical e horizontal [11]; (3) lagos: número de lagos e posicionamento dos seus centros [6]; (4) finais de linha: número e posicionamento dos finais de linha de um caractere [6, 11]; (5) junções: número e posicionamento das junções do caractere [6, 11].

Durante a análise dessas características, vão sendo eliminados os caracteres da base que não são semelhantes ao objeto que está sendo comparado. Ao final, podem restar zero, um ou mais de um caractere da base. Se restar zero

ou um caractere, então o reconhecimento atingiu o seu fim. Se restar mais de um caractere, então entra em ação a fase de Comparação de Modelos Probabilísticos. Esta fase compara os modelos dos caracteres da base que restaram com o objeto que está sendo comparado. Os modelos da base são ponderados, durante a fase de treinamento, de acordo com a freqüência dos valores que cada posição do modelo pode assumir. Ao final da comparação, para cada caractere restante da base, temos uma probabilidade de o objeto que está sendo comparado ser aquele caractere. O caractere da base com a maior probabilidade é escolhido como o correto.

3. Resultados Experimentais

O experimento realizado foi baseado nos testes feitos em [1], que utilizava imagens em escala de cinza contendo todos os caracteres da tabela ASCII, formatados em 100 fontes diferentes e 10 tamanhos entre 5 e 14. Foram feitas algumas modificações no experimento apresentado em [1]. Foram utilizadas 168 imagens em escala de cinza, cada imagem continha todas as letras do alfabeto inglês, maiúsculas e minúsculas, e também dígitos de 0 a 9. Os textos nas imagens foram formatados em 24 tamanhos de 6 a 60, e em 7 tipos de fonte: *Arial*, *Avant Garde*, *Bitstream Vera Sans*, *Courier 10 Pitch*, *Helvetica*, *Times*, *Times New Roman*, que são as fontes mais utilizadas em ambientes Desktop e Web [7].

A comparação foi feita entre o *imagocr*, o *gocr* e o *ocrad*. Não foram inseridos no experimento os sistemas anteriormente mencionados: (1) *calera*, por ser proprietário; (2) o sistema descrito em [12, 13], por não ser disponível para download gratuito; e (3) *tesseract*, por somente aceitar como entrada imagens bitonais.

A Fig. 3 mostra um exemplo de imagem utilizada no experimento, bem como os resultados correspondentes para cada um dos sistemas comparados. A Fig. 4 apresenta, para os três sistemas testados, a taxa de reconhecimento média em cada tamanho de fonte. A avaliação completa para cada sistema está disponível na página do grupo⁴.

⁴ www.imago.ufpr.br/imagocr

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
 imagocr: THE QU7cK BROWN FOX JuMPS OVER THE _ZY DOG
 gocr: THE au ch BROWN Fox Jums OVER THE _DOG
 ocrad: THE OUICK BROWN FOX_UMP_ OV_R THE _ DOC

Figura 3. Exemplo de imagem usada no experimento e saídas correspondentes dos três sistemas. De cima para baixo: imagem original, fonte Arial tamanho 14; e saídas do imagocr, gocr e ocrad, respectivamente.

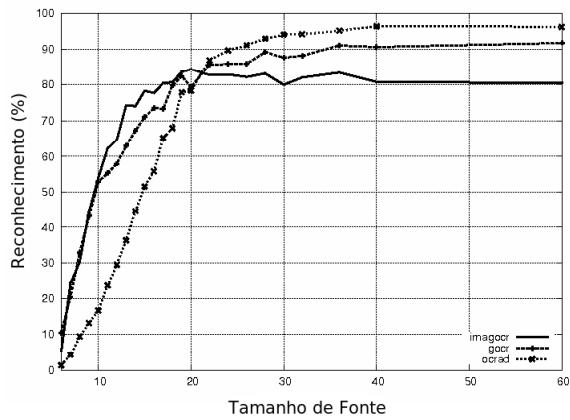


Figura 4. Comparação dos sistemas de OCR com a taxa de reconhecimento média em cada tamanho de fonte.

O sistema proposto atinge um desempenho equivalente aos outros sistemas utilizando uma base de dados menor. O tempo médio de execução do sistema, para cada imagem, foi abaixo de 0.3 segundos. Também é possível perceber que o *imagocr* é mais preciso quando o tamanho varia de 9 a 22. Segundo [7], os tamanhos de fonte mais utilizados em ambientes *Desktop* variam de 10 a 12, e na *Web* de 12 a 14. Por isso, o treinamento da base de dados foi realizado somente para estes tamanhos de fonte. As etapas de segmentação de caracteres, Análise Estrutural e Comparação de Modelos Probabilísticos também foram construídas utilizando caracteres pequenos. A consequência disso é que, para tamanhos maiores, a taxa de reconhecimento se estabiliza, pois, por mais que fontes maiores permitam a extração de mais características, o método limita-se apenas a características inerentes a fontes menores.

4. Conclusão

Através de um conjunto pequeno de características e um mecanismo simples de comparação de modelos proba-

bilísticos, é possível implementar a base, ou núcleo, de um sistema OCR. Isso se deve ao uso de métodos híbridos de reconhecimento, que exploram as principais propriedades de métodos distintos. Utilizando um método híbrido, também é possível diminuir consideravelmente o número de amostras da fase de treinamento.

Como apresentado na seção anterior, o sistema proposto destaca-se no reconhecimento de fontes de tamanhos de 9 a 22, que são os tamanhos mais usados em computadores pessoais hoje em dia, seja para aplicativos *Desktop* ou *Web*. Devido às suas etapas de pré-processamento e segmentação, o sistema proposto limita-se apenas a imagens com fundo bem comportado. Entretanto, o sistema pode ser utilizado com eficiência no reconhecimento de textos em aplicativos de maneira a auxiliar pessoas com necessidades especiais no uso do computador.

Referências

- [1] H. S. Baird and R. Fossey. A 100-font classifier. *Proc. IC-DAR*, pages 332–340, 1991.
- [2] M. Bokser. Omnidocument technologies. *Proceedings of the IEEE*, 80(7):1066–1078, 1992.
- [3] R. G. Casey and E. Lecolinet. A survey of methods and strategies in character segmentation. *IEEE Trans. PAMI*, 18(7):690–706, 1996.
- [4] R. C. Gonzalez and R. E. Woods. *Processamento de Imagens Digitais*. Edgard Blücher, 1 edition, 2000.
- [5] IMAGO. Projeto linux acessível. Disponível em: http://www.imago.ufpr.br/pt_linuxacessivel.html. Acesso em: 25 ago 08.
- [6] O. Martinsky. Algorithmic and mathematical principles of automatic number plate recognition systems. Master's thesis, Brno University of Technology, 2007.
- [7] E. R. Oliveira. Avaliação ergonômica de interfaces da scielo - scientific electronic library online. Master's thesis, Universidade Federal de Santa Catarina, 2001.
- [8] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9(1):62–66, 1979.
- [9] J. R. Parker. *Algorithms for Image Processing and Computer Vision*. John Wiley and Sons, 1997.
- [10] R. Smith. An overview of the tesseract ocr engine. *Proc. IC-DAR*, pages 629–633, 2007.
- [11] O. D. Trier, A. K. Jain, and T. Taxt. Feature extraction methods for character recognition - a survey. *Pattern Recognition*, 29(4):641–662, 1996.
- [12] S. Wachenfeld, H. U. Klein, S. Fleischer, and X. Jiang. Segmentation of very low resolution screen-rendered text. *Proc. of Int'l. Conf. Document Analysis and Recognition*, pages 1153–1157, 2007.
- [13] S. Wachenfeld, H. U. Klein, and X. Jiang. Recognition of screen-rendered text. *Pattern Recognition*, pages 1086–1089, 2006.

Aquisição de Dados para um Sistema de Captura de Movimentos

Carolina A. S. Bigonha[†], David L. Flam (orientador)[†]

João Victor B. Gomide (orientador)[§], Arnaldo de A. Araújo (orientador)[†]

[†] Departamento de Ciência da Computação, [§]Escola de Belas Artes – Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627, Pampulha, Belo Horizonte, Minas Gerais, Brasil
{carolb, david, arnaldo}@dcc.ufmg.br, jvbgomide@gmail.com

Resumo

O presente trabalho aborda a localização e a identificação de marcadores em um software de aquisição de dados para um sistema de captura de movimentos. Utilizar-se-á a identificação, por meio de técnicas de detecção de quinas, de marcadores passivos que refletem infravermelho. Uma vez realizada a captura de dados proposta e, ainda, a triangulação e o rastreamento dos pontos, será possível realizar a animação de personagens.

1. Introdução

1.1. Captura de Movimento

A captura e a análise do movimento humano são áreas de pesquisa relevantes devido ao vasto campo de aplicação e à complexidade inerente.

As potenciais aplicações dessa área são divididas em três conjuntos, segundo Moeslund *et al.* [5]: monitoramento – quando um ou mais alvos são rastreados e monitorados ao longo do tempo –, controle – correspondente à construção de interface para jogos, ambientes virtuais e, ainda, para fazer animação – e análise – que engloba a análise detalhada dos movimentos do ponto de vista clínico, permitindo realizar diagnósticos de enfermidades de pacientes ou melhorar a performance de atletas.

Pode-se dizer que o presente artigo aborda a segunda aplicação por ter como finalidade usar a captura de movimentos para fazer animação de personagens.

Os sistemas óticos de captura de movimento são divididos em, basicamente, dois subsistemas: o de detecção – hardware – e o de processamento – software.

A detecção pode ser ativa ou passiva. Na ativa, em um ambiente pré-definido e controlado, dispositivos são acoplados tanto ao alvo da detecção quanto aos arredores, que transmitem ou recebem sinais gerados, respectivamente.

Tais mecanismos de emissão e leitura de sinais facilitam a identificação e a localização dos pontos-chave em uma cena.

A detecção passiva, por sua vez, não requer vestimentas munidas de equipamentos com processamento interno: a captura pode ocorrer até mesmo a partir de seqüências de vídeo comuns. Os sensores passivos são baseados em fontes “naturais” de luz, e.g., o espectro visível ou outras ondas eletromagnéticas [3]. Portanto, quando se fala em marcadores passivos, refere-se a objetos ou marcações que, simplesmente, facilitam a segmentação e, logo, a identificação de pontos. Esses marcadores são diferentes dos citados no contexto ativo, já que os últimos respondem a estímulos externos.

Sistemas de captura do tipo passivo são preferidos por serem uma boa alternativa em termos de custo, facilidade de uso e adaptação. Outro importante aspecto é que eles não são intrusivos, permitindo um comportamento humano natural perante câmeras.

Para este projeto, preferiu-se a detecção passiva feita por marcadores, da mesma categoria, que refletem infravermelho.

1.2. Os Softwares 3D

Em softwares para modelagem 3D, o personagem virtual é feito usando-se estruturas articuladas. Os modelos articulados, denominados *skeletons*, representam de maneira realista o movimento humano: são conjuntos de objetos rígidos, também conhecidos como *bones*, conectados por articulações – ou juntas –, que permitem o movimento.

Convém ressaltar que tais articulações, ou juntas, obedecem a uma hierarquia semelhante à de uma árvore. Dessa maneira, apenas a primeira articulação da estrutura precisa ser posicionada no espaço: as demais são posicionadas de acordo com os ângulos relativos entre os pontos definidos anteriormente.

A raiz da estrutura, conforme mostra a Fig. 1, é localizada no centro do personagem. Cada ponto relevante – e.g.

cotovelo, ombro, joelho – pertence a um nível da hierarquia e depende diretamente do ponto pertencente ao nível anterior.

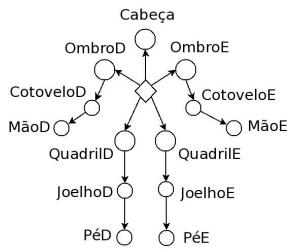


Figura 1. Hierarquia de um personagem.

Há uma série de formatos de arquivos que armazenam os dados extraídos de uma cena durante uma captura de movimento. No sistema proposto utilizar-se-á um arquivo com o padrão *Biovision Hierarchy* (BVH) que armazena os movimentos detectados, organizando os dados em duas grandes seções: hierarquia e movimento [4].

Na seção correspondente à hierarquia estabelece-se a relação entre os vértices, e, na correspondente aos movimentos, armazena-se as transformações de cada ponto de acordo com o fluxo de tempo.

Escolheu-se esse formato dada sua simplicidade e compatibilidade com os mais conhecidos *softwares* de modelagem 3D, como o *3D Studio Max*, *Blender3D*, *Lightwave* e *Maya*.

2. Metodologia

A metodologia adotada pode ser divida nas seguintes etapas: preparação das câmeras e do ambiente, localização e identificação dos marcadores, rastreamento dos pontos, reconstrução tridimensional, e, por último, manipulação dos dados a fim de se obter a saída esperada.

Cada uma dessas etapas será explicada adiante e está representada no diagrama da Fig. 2.

2.1. Decisões de Projeto

Para realizar a captura de dados, pretende-se trabalhar com câmeras digitais de vídeo preto e branco com resolução de 1280x1024 pixels. Ao trabalhar com mais de uma câmera, uma representação tridimensional pode ser obtida.

A fim de poupar o processamento de localização dos pontos, propôs-se que cada marcador passivo possuísse um desenho único, distingüível dos demais. Dessa maneira, tornar-se-ia fácil encontrar um mesmo ponto tanto em dois quadros quanto nos vídeos de duas câmeras distintas.

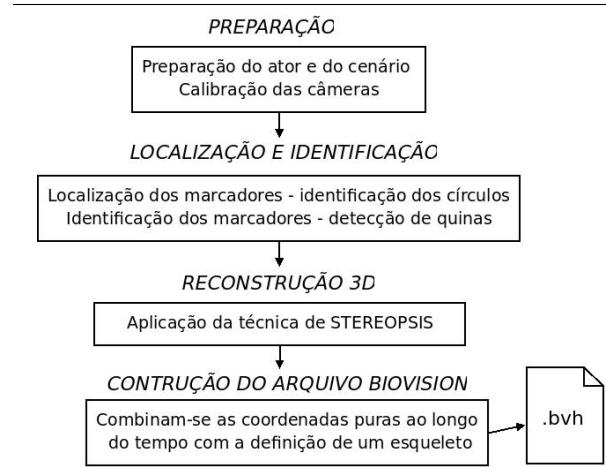


Figura 2. As etapas do sistema.

Aproveitando-se a alta resolução da câmera disponível para a captura, optou-se por um desenho que pudesse ser identificado por um procedimento de detecção de quinas. Essa técnica foi escolhida por ser relativamente simples e atender bem ao propósito.

Cada articulação possuirá um marcador com um certo número de quinas, permitindo-se, assim, a distinção entre cada ponto, conforme exemplifica a Fig. 3. Os marcadores passivos serão posicionados no corpo do ator de acordo com uma tabela, que indicaria qual marcador pertence a qual junta.

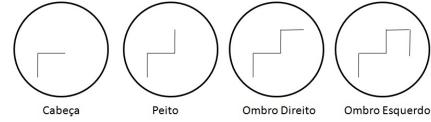


Figura 3. Identificação dos marcadores.

Para completar o material, têm-se disponíveis, ainda, emissores de luz na faixa do infravermelho, filtros para o mesmo comprimento de onda e marcadores passivos que refletem nessa mesma faixa. Tais marcadores, têm formato circular, possuem identificação e são posicionados estratégicamente no corpo do ator, em pontos-chave como cabeça, ombros e joelhos.

A implementação dos algoritmos propostos foi feita usando a biblioteca *Open Source Computer Vision Library*, ou OpenCV.

2.2. Preparação

Antes de realizar a aquisição de dados, há uma série de condições – que envolvem tanto o ambiente quanto o

ator – que devem ser satisfeitas para que se tenha uma interpretação correta da cena a ser capturada.

Primeiramente, o ambiente deve ter uma iluminação constante – no caso, infravermelha – e um fundo estático e uniforme. As câmeras devem permanecer paradas, posicionadas sobre tripés. O alvo, ou ator, deve usar uma vestimenta adequada e deve ter os marcadores descritos anteriormente, posicionados em certos pontos de seu corpo.

Satisfitas tais condições, deve-se ajustar o sistema de acordo com as características particulares do cenário e calibrar as câmeras.

A chamada 'inicialização do modelo' é mais uma etapa crucial na preparação do sistema [5]. A maioria dos sistemas existentes adota a pose inicial do objeto como uma pose específica e pré-definida. Outras alternativas podem ser o operador do sistema especificar a pose inicial como parâmetro ou verificar o posicionamento do alvo em todas ou alguns quadros. Nesse último caso, o sistema simplesmente faz a relação entre eles.

O presente trabalho procura reduzir ou até eliminar essa fase de inicialização, já que os marcadores serão identificáveis a qualquer momento. O objetivo é que apenas ajustes opcionais precisem ser feitos pelo operador, deixando, assim, o programa mais robusto.

2.3. Localização e Identificação dos Marcadores

A reflexão do infravermelho pelos marcadores é captada pela câmera na forma de um borrão branco. Em vista disso, a realização de um pré-processamento agiliza a detecção significativamente.

Aplica-se, em cada quadro, uma limiarização binária – fazendo com que todos os pixels com valor menor que o limiar sejam transformados em pretos, e os pixels com valor maior ou igual ao limiar sejam transformados em brancos. Em seguida, realiza-se uma suavização de Gauss para que pequenos detalhes e distorções sejam removidos e para que se obtenham círculos mais bem delimitados. Por fim, aplica-se um algoritmo para detecção de bordas utilizando-se a técnica de Canny, que, embora tenha maior complexidade, produz resultados melhores [2].

Nesse ponto do processamento, tem-se uma imagem que contém somente o contorno dos círculos que representam os marcadores. Aplica-se, portanto, o algoritmo de detecção de círculos proposto em [1]. Em poucas palavras, tal algoritmo constrói um vetor com todos os pixels da borda e avalia, para cada conjunto de quatro pixels, se eles formam um círculo. Caso afirmativo, verifica-se se o círculo obedece aos parâmetros estabelecidos, como, por exemplo, diâmetro máximo. A saída desse algoritmo são as coordenadas do centro dos marcadores e seus respectivos raios.

Uma vez localizados os marcadores, passa-se para a segunda fase, que consiste na identificação dos mesmos.

Realiza-se a busca por quinas somente nas regiões que correspondem aos pontos encontrados. Pode-se, assim, identificar cada marcador.

Após o término dessa etapa, obtém-se as coordenadas do centróide dos marcadores nos planos que representam a imagem de cada câmera. Assim, bastará fazer a relação de posição dos marcadores entre os quadros do vídeo. Técnicas avançadas de *tracking* são dispensadas pelo fato de sempre se conhecer as coordenadas de cada um dos marcadores, mas serão utilizadas para predição do movimento em casos que ocorram oclusão de pontos.

2.4. Reconstrução 3D

Unir as informações gravadas por mais de uma câmera, explorando as disparidades entre elas, permite que se obtenha um senso de profundidade [6]. Para transformar os pontos independentes de cada câmera do sistema em coordenadas tridimensionais, a chamada *stereopsis* realiza dois processos: a fusão das características observadas de duas ou mais câmeras e a reconstrução da imagem tridimensional.

O uso dessa técnica permitirá que se encontrem as coordenadas x , y e z de cada marcador.

2.5. Preparação dos Dados para Animação

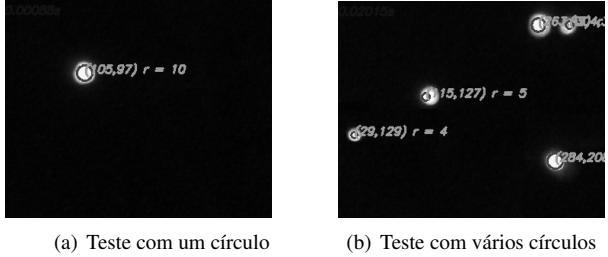
Os dados puros, extraídos da cena, até então, são chamados de dados de translação global [4]. Eles representam a posição de cada marcador em relação à origem global, sem incluir nenhuma hierarquia ou definição de esqueleto.

Embora trabalhar com os próprios arquivos translacionais globais seja possível e, ainda, ofereça uma maior liberdade quanto a complexidade estrutural dos personagens, para o projeto proposto, é interessante transformar as coordenadas cartesianas coletadas em um formato hierárquico. Além do fato de que trabalhar-se-á com personagens humanóides, as formas de armazenamento que mantêm uma relação dos pontos com um esqueleto deixam a configuração da estrutura do personagem mais independente do modelador 3D e mais próximo do *software* que gera os dados.

Uma vez realizada a tradução das coordenadas cartesianas puras para o formato .bvh, descrito anteriormente, basta importar o arquivo em um *software* 3D e aplicá-lo a um personagem.

3. Resultados Parciais

Até o presente momento, desenvolveu-se a parte de localização e identificação dos marcadores.



(a) Teste com um círculo (b) Teste com vários círculos

Figura 4. Localização dos marcadores.



(a) Marcador com quina desenhada (b) Detecção de quinas

Figura 5. Identificação dos marcadores.

Os testes apresentados a seguir, foram feitos via simulação da recepção de infravermelho por uma câmera por meio de sinal de controle remoto.

As Figuras 4(a) e 4(b) mostram o centro e o raio dos marcadores fictícios – os borrões brancos da imagem – encontrados como resultado da detecção de círculos aplicada.

As imagens da Fig. 5 ilustram o desempenho da detecção de quinas. Em 5(a) tem-se uma imagem sintética que simula o desenho de um dos marcadores. Em 5(b) tem-se a saída do algoritmo empregado.

Por último, pode-se observar, por meio dos gráficos a seguir, o desempenho dos algoritmos de detecção de círculos – Fig. 6 – e de quinas – Fig. 7 – em relação ao tempo médio de processamento gasto. Para cada um dos algoritmos, variou-se os seguintes parâmetros: no primeiro, o número de pontos e, no segundo, o número de quinas.

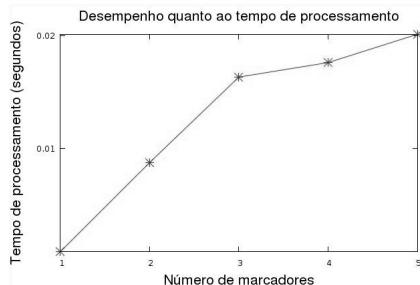


Figura 6. Tempo x Número de marcadores.

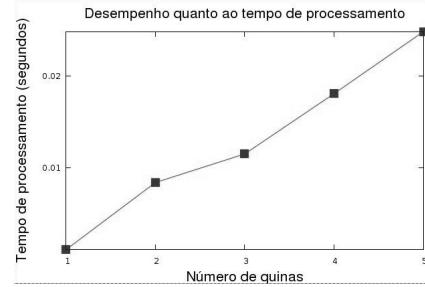


Figura 7. Tempo x Número de quinas.

4. Conclusão e Trabalhos Futuros

Inicialmente, o objetivo é implementar todo o *software* proposto e testá-lo para a animação de personagens. Uma vez alcançado esse objetivo pretende-se comparar o sistema desenvolvido com outros disponíveis no mercado. A comparação em termos de velocidade, qualidade de resultado e custo será de suma importância para a melhoria da qualidade do sistema.

Acredita-se que a utilização de marcadores passivos identificáveis será de suma importância para a construção de um sistema de captura rápido e de baixo custo.

Pode-se dizer que o aumento da eficiência se relaciona diretamente com a diminuição do processamento gasto com o rastreamento dos pontos, que é uma das etapas mais custosas do processo de captura. Se cada articulação é identificável em qualquer quadro, não há necessidade de se fazer cálculos complicados a cerca do movimento dos marcadores.

Os autores agradecem ao CNPq e à FAPEMIG pelo suporte financeiro deste trabalho.

Referências

- [1] T. C. Chen and K. L. Chung. An efficient randomized algorithm for detecting circles. *Computer Vision and Image Understanding*, (83):172–191, April 2001.
- [2] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Pearson Prentice Hall, 3rd edition, 2008.
- [3] Kakadiaris and C. Barrón. *The Handbook of Mathematical Models in Computer Vision*. Springer, 2008.
- [4] A. Menache. *Understanding Motion Capture for Computer Animation and Video Games*. Morgan Kaufmann; 1st edition, 1999.
- [5] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, (104):90–126, 2006.
- [6] J. Ponce and D. A. Forsyth. *Computer Vision - A Modern Approach*. Prentice Hall, 2002.

SMALL GPS – Sistema de Mapeamento de Linhas de Ônibus e Localização do Usuário via GPS pelo Celular

Carlos Henrique Macchi Gasparetto¹, Isabel Harb Manssour²(orientadora)

PUCRS – FACIN (Faculdade de Informática)

Av. Ipiranga, 6681 - Prédio 32 - 90619-900 - Porto Alegre/RS - Brasil

cgasparetto@gmail.com¹, isabel.manssour@pucrs.br²

Resumo

A cada dia que passa aumenta a capacidade de processamento, armazenamento e as funcionalidades disponíveis em palmtops e celulares. Agendas, jogos e planilhas são exemplos das inúmeras aplicações existentes para esses dispositivos. No entanto, outras aplicações utilizadas em computadores pessoais também poderiam ser portadas para esses aparelhos. O objetivo deste trabalho é apresentar o projeto e desenvolvimento do protótipo de uma aplicação de localização geográfica para dispositivos móveis, que visa fornecer a visualização de um mapa, indicando através de GPS a localização do usuário e exibindo quais linhas de ônibus passam nas proximidades daquele local. Além disso, é possível ter acesso a informações de itinerário e horários de cada linha, e de pontos de interesse por onde elas passam.

1. Introdução

Portabilidade é uma tendência nesse início de século. Cada vez mais se destacam o uso de celulares e palmtops que aumentam rapidamente seu poder de processamento, armazenamento e suas funcionalidades. Há uma década, o celular era utilizado basicamente para falar, contando, no máximo, com uma agenda de pequena capacidade. Hoje os celulares possuem acesso à Internet, tocadores de rádio e música, câmeras digitais e muitas outras funções. Dentro deste contexto, procura-se cada vez mais portar as funcionalidades existentes em um PC para esses aparelhos.

Estudos realizados apontam que o número de usuários de celulares já representa um terço da população total do planeta e a cobertura dos serviços chegará a 90% do globo terrestre até 2010. No Brasil, segundo a Anatel, existem cerca de 133 milhões de celulares em uso, para uma população de 187 milhões de pessoas [1]. Considerando o grande número de usuários, a capacidade dos dispositivos e algumas funcionalidades ainda pouco exploradas, surgiu a

motivação para o desenvolvimento deste trabalho. Em grandes cidades é comum as pessoas irem a locais com os quais não estão familiarizadas, e não saberem que transporte pegar. Nestes casos, surge a necessidade de pedir informação ou recorrer a um mapa. Mas, normalmente, estes são grandes e pouco práticos de carregar, e a pessoa pode não identificar onde está. Então, o melhor é disponibilizar estas informações de forma automática e interativa no celular.

O objetivo deste trabalho é apresentar o projeto e o desenvolvimento de um sistema de localização com GPS e mapeamento de linhas de ônibus pelo celular. O protótipo implementado fornece a visualização de um mapa, indicando através de GPS a localização atual do usuário sobre o mapa e exibindo quais linhas de ônibus passam nas proximidades daquele local, possibilitando visualizar informações como itinerário e horários de cada linha, além de pontos de interesse por onde elas passam. Este sistema utiliza tecnologias gratuitas, não depende de serviços das operadoras de telefonia móvel e considera que a maioria dos aparelhos estará equipada com GPS até 2011.

Na Seção 2 são apresentados alguns trabalhos relacionados. O SMALL GPS está descrito na Seção 3 e as conclusões e trabalhos futuros estão na Seção 4.

2. Trabalhos Relacionados

Atualmente existem alguns serviços comerciais de mapas para celulares, porém a maioria é paga, restrita a determinadas operadoras e poucos estão disponíveis no Brasil. O Guia Quatro Rodas [2], por exemplo, disponibiliza informações sobre locais de interesse para o usuário, como hotéis e restaurantes, mas não possui sistema de localização. É compatível com vários modelos de aparelhos celulares, mas só está disponível para clientes de uma operadora, mediante cobrança de tarifa. O *Google Maps Mobile* [4] é a versão para dispositivos móveis do serviço de mapas do Google. É gratuito e permite traçar rotas, pesquisar endereços e exibir imagens reais de satélite. Mas, ainda não tem

integração com um sistema de localização, nem exibição de linhas de ônibus. Vivo Encontra utiliza o método de triangulação de antenas para permitir verificar a posição de outros celulares [3]. Precisa estabelecer uma conexão de dados com a plataforma, que enquanto estiver ativa não permite receber outra chamada. Esses serviços são cobrados, restritos aos assinantes da operadora e a poucas cidades.

Alguns trabalhos acadêmicos também foram desenvolvidos nesta área. Um deles [5] descreve em detalhes a utilização da API (*Application Program Interface*) *Location* (especificação JSR-179) para geolocalização através de dispositivos sem fio. Outro [6], teve como foco o uso de GPS e celular na agricultura. Foi elaborado um sistema que identifica se existe sinal de alerta sobre doenças na região onde o usuário está, e se as áreas afetadas podem comprometer a sua área de plantio.

O “Borboleta Sistema Móvel para Telemedicina” [7] é direcionado a computadores de mão. Funciona como um prontuário eletrônico e fornece um mapa para localização das casas dos pacientes, mas, não usa GPS e possui uma baixa qualidade na visualização do mapa. Um mapa para ser usado na área de turismo em uma região do território chinês também foi elaborado [8]. O trabalho não usa GPS, foi implementado em J2ME e usa a *Mobile SVG*, uma especificação limitada do SVG padrão, e a *TinyLineSVG*, uma API desenvolvida para a manipulação do *Mobile SVG*.

O *Mobile GIS* [9] não usa GPS e tem o objetivo de permitir a visualização de informações geográficas em dispositivos móveis. Estas informações são armazenadas em um servidor e representadas em *Geography Markup Language* (GML) [10], que é baseada em XML e voltada à representação de informação geográfica. O arquivo GML é disponibilizado em um servidor, transformado em SVG e enviado ao cliente.

3. SMALL GPS

SMALL GPS é a sigla de Sistema de MApeamento de Linhas de ônibus e Localização do usuário através de **GPS** pelo celular, fazendo também analogia com o tamanho dos dispositivos envolvidos (*small*). Este sistema permite visualizar um mapa com as ruas próximas ao local onde o usuário está, o itinerário das linhas de ônibus e os pontos de interesse da região, tais como restaurantes, parques e hospitais. Portanto, o mapa é útil para moradores da cidade e para turistas.

Para o protótipo do programa, foi escolhida a cidade de Porto Alegre, que possui uma grande malha viária e um dos melhores sistemas de transporte coletivo do Brasil [11]. Além disso, a Companhia Carris Porto-

Alegrense, que concentra 32% do fluxo diário de passageiros [12], está com um sistema em fase experimental que mostra no site da companhia a localização dos ônibus de determinadas linhas em tempo real, via GPS [12]. A expansão deste serviço, que é uma das pretensões da empresa, poderá ser incorporada em futuras versões deste trabalho.

3.1. Tecnologias Utilizadas

O sistema foi implementado na linguagem de programação Java. Para a camada cliente também foram utilizadas as especificações *Mobile Information Device Profile* (MIDP) 2.0 e *Connected Limited Device Configuration* (CLDC) 1.1, que são o núcleo do J2ME. Ainda foram necessárias as APIs *Location* (JSR-179), para localização geográfica, e a *Scalable 2D Vector Graphics* (JSR-229) para visualização dos arquivos SVG [13]. Para a camada servidor foi utilizada a plataforma J2EE (*Java 2 Enterprise Edition*) e a interface *JavaServer Pages* (JSP) [14].

O ambiente de desenvolvimento escolhido foi o Eclipse [15], versão 3.2, juntamente com dois *plugins*. Um é o *Eclipse ME* [16], versão 1.6.7, responsável pela integração do Eclipse com kits de desenvolvimento para dispositivos móveis. Neste trabalho, foi escolhido o kit *Sun Java Wireless Toolkit* [17], versão 2.5. O outro *plugin* é o *Sysdeo Tomcat Plugin* [18], versão 3.2, que auxilia na integração com o servidor de aplicação *Apache Tomcat*, usado na camada servidor da aplicação para facilitar a inicialização do servidor e a visualização de *logs*, mensagens de erro e depuração.

3.2. Armazenamento e Manipulação do Mapa

Como formato de exibição dos mapas, optou-se pelo SVG [19], que consiste no armazenamento de formas geométricas e linhas em XML (*Extensible Markup Language*). As vantagens do SVG para utilização em dispositivos móveis são: arquivo de tamanho pequeno; permite fazer ampliação (*zoom*) sem perder qualidade; formato de armazenamento em XML permite edição e montagem dinâmica através de dados geográficos armazenados em um banco de dados remoto. Neste trabalho, o SVG é decodificado no celular pelo *parser* da API JSR-229. Para o protótipo do sistema algumas áreas do mapa foram desenhadas no editor de arquivos SVG gratuito *Sketsa* [20].

Como um arquivo com um mapa inteiro de uma cidade é muito grande, considerando a rede instável e a pequena memória dos celulares atuais, o mapa foi segmentado em quadrantes, como em listas telefônicas. Assim, em uma visualização são carregados nove

quadrantes, um central e oito em volta deste, fazendo com que o tamanho dos dados de visualização fique leve e possa ser desenhado pelo dispositivo. Os quadrantes têm escala aproximada de 300 metros de largura e de altura, e as linhas de ônibus cujo itinerário passa dentro de um dos nove quadrantes visualizados, são consideradas próximas do usuário. Isso simplifica a localização das linhas e diminui o tempo de execução.

O armazenamento dos dados é feito no banco de dados (BD) Oracle Express 10g [21]. Cada quadrante possui informações de ruas, pontos de interesse, etc, independente dos demais quadrantes. Os dados cartográficos dos quadrantes do mapa são armazenados no BD como primitivas (por exemplo, retas e curvas) já no formato SVG. Na recuperação do mapa do BD, obtém-se o quadrante central e seus oito quadrantes vizinhos (Figura 1). Os dados dos nove quadrantes são unidos em um único arquivo, no qual é inserido o cabeçalho SVG, que é retornado à aplicação através da rede. Outros dados como as listas das ruas por onde passam as linhas de ônibus e os seus horários, são armazenados em tabelas do BD do sistema.



Figura 1 – Exemplo divisão da tela em quadrantes

A localização do usuário, a partir da qual são buscados os quadrantes, é obtida através de GPS. A interação do SMALL GPS com o GPS do celular ocorre através da API *Location*, que encapsula este processo. Neste trabalho foi utilizado um gerador de eventos externos que acompanha o simulador e gera coordenadas como em um GPS real.

3.3. Arquitetura

A arquitetura escolhida foi a “cliente-servidor”, que se mostra vantajosa para uma aplicação para celular, aparelho com poder limitado de armazenamento e processamento. A aplicação cliente é instalada em um celular, e a servidor em um computador remoto que contém o BD. Este banco é acessado através de uma aplicação Java/JSP, que é invocada pela aplicação cliente. O servidor recebe alguns parâmetros de acordo com as opções escolhidas pelo usuário e a posição do GPS, que são utilizados na construção das pesquisas (*queries*). A Figura 2 ilustra este processo.

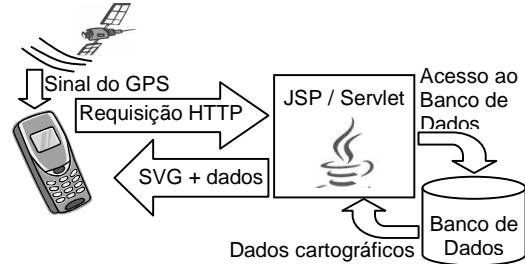


Figura 2 – Diagrama do ciclo de execução

O fluxo de execução inicia quando o usuário abre a aplicação cliente no seu celular. O aparelho recebe o sinal do GPS e, a partir da posição detectada, é feita uma requisição sobre protocolo HTTP, na qual as coordenadas geográficas da posição do usuário são passadas como parâmetros ao servidor. O servidor processa esta requisição e retorna os dados sobre as linhas de ônibus e um arquivo SVG com o mapa da região correspondente. No cliente, o arquivo é lido e os objetos são desenhados na tela do celular, finalizando o ciclo. No caso de interações do usuário, tais como deslocamento do mapa (*pan*) e seleção de exibição do traçado de uma linha de ônibus, é feita uma requisição, indicando como parâmetros, respectivamente, a direção do *pan* ou a linha de ônibus selecionada.

O servidor é responsável por receber as requisições remotas de n aplicações clientes, instaladas em vários aparelhos celulares. A partir do tipo de requisição, ele manipula os dados cartográficos através de pesquisas ao BD, e transforma-as em dados manipuláveis pela aplicação cliente. Foi usado o modelo MVC (*Model-View-Controller*) na implementação [14]. A camada *view* serve para interação com a camada cliente, tarefa desempenhada por um arquivo JSP que espera a conexão de um cliente. Quando esta ocorre, a camada *controller* manipula os dados obtidos pela camada *view* e repassa-os à camada *model*, que estrutura uma pesquisa que traz os dados a serem passados ao usuário.

Os resultados da pesquisa são estruturados em um arquivo SVG que recebe um nome baseado em um número randômico, gerado de forma a nunca se repetir. Os arquivos SVG são excluídos periodicamente com uma rotina automática, para liberar espaço em disco. Outros dados, como tabelas de horários, são retornados à camada *controller* que devolve os resultados para a camada *view* que os envia ao cliente.

3.4. Funcionalidades

Após uma requisição do cliente, o mapa é montado e exibido ao usuário, inicialmente, com as ruas, ícones de locais de interesse e um ponto que representa a

localização atual do usuário (Figura 3a). Depois que o mapa é exibido, pode-se acessar o menu e optar por fazer *zoom in*. No início o mapa é apresentado com afastamento máximo, mas, após o *zoom in*, pode-se fazer *zoom out* até o nível de afastamento original.

O menu principal da aplicação possui dois itens: opções e seleção de linhas de ônibus. O primeiro permite alterar parâmetros de configuração, como as cores do mapa. O segundo possibilita selecionar uma linha de ônibus. Após esta seleção surgem três opções: itinerário por escrito, que lista todas as ruas nas quais o ônibus passa; itinerário no mapa, para exibir ou ocultar no mapa as linhas gráficas dos itinerários dos ônibus na cor predominante da empresa de transporte responsável por ela (Figura 3b); e tabela de horários. Apenas os ônibus que passam perto do local onde o usuário se encontra estarão disponíveis para seleção, ou seja, os ônibus cuja rota passe dentro do quadrante onde o usuário está ou de um dos oito quadrantes vizinhos.

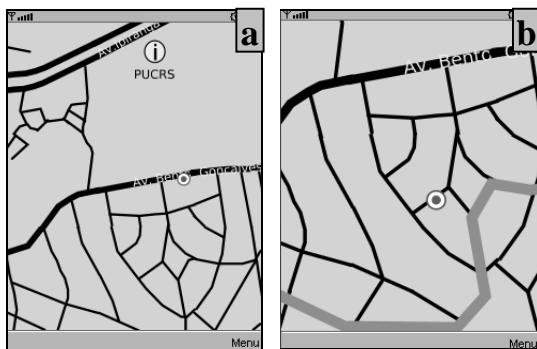


Figura 3 – Exemplo de exibição (a) inicial de um mapa e do (b) itinerário de uma linha de ônibus.

Também é possível navegar pelo mapa nas direções norte, sul, leste e oeste (*pan*). O *pan* é feito carregando novos quadrantes, assim que chegar à extremidade do mapa carregado no momento. Por exemplo, se o usuário movimentou o mapa na direção leste, a aplicação moverá uma coluna de quadrantes para a direita, carregando uma nova coluna e descartando a mais à esquerda. Para norte e sul o processo é igual, mas incluindo e removendo linhas ao invés de colunas.

4. Conclusões e Trabalhos Futuros

Como o desenvolvimento deste trabalho envolveu um conjunto de tecnologias ainda muito recentes, a aplicação resultante será mais utilizada num futuro próximo, quando as funcionalidades de GPS e gráficos vetoriais estarão disponíveis na maioria dos aparelhos celulares. Além da utilização de tecnologias gratuitas, e de não depender de serviços da operadora, exceto uma conexão com a Internet, o trabalho mostra inovação em

relação aos trabalhos acadêmicos analisados, por unir mapeamento urbano à localização via GPS, característica não encontrada em nenhum deles. Ainda, a integração de mapeamento de linhas de ônibus neste contexto mostra-se uma novidade, já que nenhum dos aplicativos analisados possui essa característica.

Está em andamento a criação de uma interface *user-friendly* para adição das informações no BD, pois isto está sendo feito diretamente através de scripts SQL. Como trabalhos futuros, pretende-se incluir o mapeamento das estações de metrô, abrangendo os usuários desse tipo de transporte. Outra adição ao sistema seria a pesquisa por ruas e locais de interesse, que consiste na inclusão de informações extras no banco de dados. Por último há a possibilidade de integração com o sistema de GPS implantado nos ônibus, que vem sendo utilizado em fase experimental em três veículos pela Companhia Carris. Se o sistema atual se expandir e se for disponibilizado para acesso público via *web services*, por exemplo, será possível estudar sua integração ao SMALL GPS.

Referências

- [1] http://www.ibge.gov.br/home/popup_popupclock.htm.
- [2] <http://guia4rodas.abril.com.br/celular>.
- [3] <http://www.vivo.com.br>.
- [4] <http://www.google.com.br/gmm/index.html>.
- [5] T. Nacif, J. Nogueira, E. Lopes. *Sistemas Baseados em Localização em um Mundo sem Fio com J2ME e API Location*. Monografia, Centro de Ciências Exatas e Tecnologia, Universidade da Amazônia, Belém, 2006.
- [6] R. Ogliari. Integração da Tecnologia GPS para o Auxílio na Tomada de Decisões. *V Congresso Brasileiro de Agroinformática*, 2005, Londrina.
- [7] P. Matsumoto, R. Miyasaki, A.P. Mota. *Borboleta Sistema Móvel para Telemedicina*. Monografia, Instituto de Matemática e Estatística, USP, São Paulo, 2005.
- [8] B. Wu; B. Xia. Mobile phone GIS based on mobile SVG. *Geoscience and Remote Sensing Symposium*, 2005.
- [9] J.C.S. Cardoso. *m-GIS: Mobile GIS*. Monografia, Faculdade de Engenharia, Universidade do Porto, 2003.
- [10] <http://www.opengeospatial.org>.
- [11] http://www.eptc.com.br/Estatistica_Transporte/PBRUTO.asp.
- [12] <http://www.carris.com.br/>.
- [13] <http://jcp.org/en/jsr/detail?id=226>.
- [14] N. Kassem. *Designing Enterprise Applications with the Java 2 Platform*. Sun Microsystems, 2000. 341 p.
- [15] <http://www.eclipse.org>.
- [16] <http://www.eclipseme.org>.
- [17] <http://java.sun.com/products/sjwtoolkit>.
- [18] <http://www.eclipsetotale.com/tomcatPlugin.html>.
- [19] <http://www.w3.org/Graphics/SVG>.
- [20] <http://www.kiyut.com/products/sketsa>.
- [21] <http://www.oracle.com/technology/xe>.

Autenticação Biométrica: um Sistema Híbrido Combinando Informações Globais e Locais de Impressões Digitais

Rubisley P. Lemes, Cristiano O. Ferreira, Olga R. P. Bellon (orientadora), Luciano Silva (orientador)

{rp103, cof06, olga, luciano}@inf.ufpr.br

Grupo IMAGO de Pesquisa*† - UFPR, Cx.P. 19092 – 81531-990 – Curitiba-PR

Resumo

Este trabalho apresenta um sistema automático para autenticação através de impressões digitais. A maioria dos métodos de reconhecimento de impressões digitais utilizam apenas minúcias, que são características locais altamente discriminantes. Já os métodos globais, que utilizam núcleo, delta e o padrão das cristas, são pouco utilizados para o reconhecimento, sendo usados principalmente para a classificação de digitais. Portanto, estes métodos locais ou globais acabam não tirando vantagem da rica informação contida no nível de detalhamento um do outro. Visando resolver as limitações destas duas categorias de algoritmos, propomos um sistema híbrido que utiliza dois diferentes métodos de reconhecimento de digitais: (1) alinhamento de imagens e (2) textura orientada. Estas duas abordagens combinam informações globais e locais das impressões digitais para obter uma melhor relação entre tempo de resposta e taxa de aceitação. Geramos inúmeros testes em diferentes bases de dados para provar a eficiência e precisão do sistema proposto. Estes experimentos também confirmaram a robustez do sistema em imagens reais de baixa qualidade.

1. Introdução

Uma grande parte dos trabalhos relacionados a impressões digitais empregam apenas minúcias para realizar o reconhecimento [5, 11, 13]. Tais métodos são altamente dependentes de resultados obtidos na fase de extração de minúcias e da qualidade do sensor. Normalmente, as imagens de entrada podem apresentar um número insuficiente de minúcias se o sensor captura somente uma pequena região da digital. Por isso, estudos recentes que destacam a utilização de sistemas híbridos têm apresentado re-

sultados promissores quando características globais (e.g. padrão das cristas) e locais (e.g. minúcias) são combinadas para o processo de reconhecimento [6, 7, 12]. Entretanto, nenhum destes trabalhos apresenta um sistema atuante no modo de identificação.

Portanto, neste trabalho, propomos um sistema híbrido de autenticação de impressões digitais, que utiliza tanto informações globais quanto locais. Este sistema utiliza alinhamento de imagens baseado no algoritmo *Simulated Annealing* (SA) [9] e na textura orientada das digitais pelo método *FingerCode* (FC) [6].

O sistema proposto possui três módulos: (1) processamento de imagens e geração dos moldes (*templates*), (2) identificação e (3) verificação. No primeiro módulo, características são extraídas das impressões digitais para gerar o molde da digital. No estágio de identificação, procura-se na base de dados as digitais mais parecidas com a digital de entrada. Em seguida, a verificação testa se uma destas melhores amostras é referente ao indivíduo da impressão digital de entrada. Portanto, o sistema proposto de autenticação emprega uma abordagem para identificação usando o FC e outra para verificação com o SA.

2. Processamento de Impressões Digitais

O módulo proposto para o processamento das imagens de impressões digitais possui os seguintes estágios: (1) pré-processamento, (2) segmentação, (3) correção da orientação da imagem, (4) estimação do campo direcional (CD), (5) estimação da freqüência, (6) máscara de reconstrução, (7) detecção do ponto de referência (núcleo) e (8) aplicação do Filtro de Gabor. Os passos 4, 5, 6 e 8 foram implementados como descritos em [4], por este ser um método estado-da-arte e incluir informações suficientes para replicação dos experimentos.

No primeiro estágio, filtros são aplicados para reduzir a quantidade de ruídos e melhorar a qualidade das imagens [2]. Foi usado o filtro da mediana para minimizar ou até mesmo eliminar informações indesejáveis geradas por fatores externos, tais como sujeira no dedo ou pressão ina-

* <http://www.inf.ufpr.br/imago>

† Os autores gostariam de agradecer ao CNPq, a CAPES e a FINEP pelo suporte financeiro.

dequada aplicada pelo dedo sobre o sensor. Também foi necessária a utilização da equalização local para maximizar a variação dos tons de cinza no histograma melhorando o contraste nestas imagens.

A Região de Interesse (RI), área com a impressão digital, é extraída no segundo estágio. Esta segmentação é importante para eliminar a possibilidade de se gerar falsas cristas ou minúcias no fundo da imagem ou em áreas muito ruidosas. Para realizar a segmentação, primeiramente dividi-se a imagem em blocos de tamanho 16×16 [4], e em seguida a variância dos tons de cinza é computada localmente para cada bloco. Também a variância global da imagem é estimada utilizando as variâncias locais calculadas anteriormente. Por último, todos os blocos com variância maior do que a variância global são selecionados como objeto (Fig. 1b). Isto acontece porque a impressão digital apresenta uma variância bem maior do que o fundo da imagem. Por meio da RI também é estimado um ângulo α (Fig. 1c) para corrigir a orientação da imagem no terceiro estágio (Fig. 1d). Para computar α , detecta-se dois pontos centrais na primeira e última linha da RI, i e j , respectivamente (Fig. 1b). No sétimo estágio, o algoritmo de Poincaré [10] é aplicado somente dentro da RI visando extrair o ponto de referência. A utilização deste ponto proporciona uma melhora substancial no desempenho do reconhecimento. Isso é decorrente da eliminação da necessidade da busca exaustiva por orientações e translações equivalentes entre as digitais que estão sendo comparadas, uma vez que o SA utiliza o ponto de referência como pré-alinhamento.

Para criar o molde necessário para a autenticação (informações utilizadas pelo FC + SA), primeiramente é executado um algoritmo para esqueletização [3] sobre a imagem resultante do estágio de processamento. Este algoritmo facilita a extração das minúcias e suas respectivas orientações [5]. O molde inclui todos os pontos pertencentes às cristas, às minúcias e o ponto de referência, com suas respectivas orientações. Em adição a estas informações também é criado um vetor de características referente à digital chamado *FingerCode* [6].

3. Algoritmos de Identificação e Verificação

Neste trabalho, o algoritmo de identificação é derivado do trabalho proposto em [6]. Tal trabalho apresenta um eficiente método estado-da-arte destinado ao reconhecimento de impressões digitais (*Fingercode*). Nesta abordagem de reconhecimento, a similaridade entre duas impressões digitais é aferida por meio da distância Euclidiana entre seus respectivos vetores de características referentes aos componentes de textura orientada da digital. Estes componentes são extraídos através de um banco direcional de Filtros de Gabor sobre uma região de interesse setorizada em torno do ponto de referência.

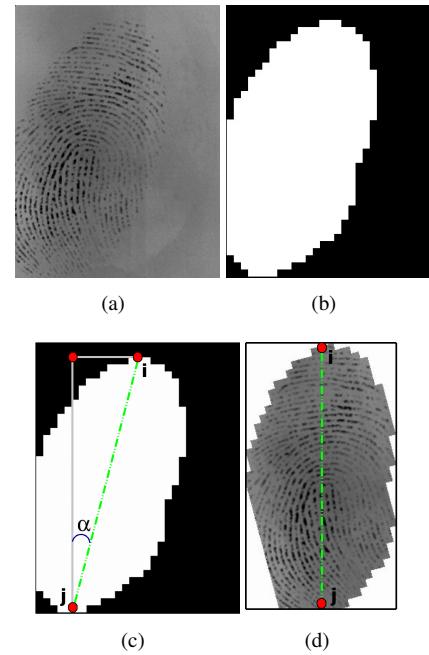


Figura 1. Pré-processamento: (a) imagem de entrada, (b) RI, (c) estimativa do ângulo de rotação e (d) imagem rotacionada.

Por outro lado, o método de verificação é o algoritmo proposto em [9]. Nesta abordagem, minúcias são combinadas às cristas para resolver limitações dos métodos baseados em características locais [5, 11, 13]. A idéia principal é gerar um alinhamento entre as cristas de duas digitais utilizando o SA [8], um algoritmo estocástico para busca local que parte de uma solução inicial e analiza soluções vizinhas que possam gerar uma melhor resposta para o problema. O pré-alinhamento (solução inicial) é executado a partir do ponto de referência da impressão digital, melhorando substancialmente a velocidade e eficiência do algoritmo. No SA, também é utilizado o CD na análise de soluções vizinhas proporcionando uma melhor distinção entre pontos sobrepostos em cristas alinhadas ou cruzadas. Por fim, são gerados dois escores: (1) taxa de pontos sobrepostos com direções equivalentes (cristas alinhadas), e (2) taxa de minúcias alinhadas com direções equivalentes. Estes dois escores são combinados gerando um escore único que define a similaridade entre as digitais. Maiores informações sobre o SA e o FC podem ser encontradas em [6, 9], pois neste artigo houve limitação de espaço para incluí-las.

4. Resultados Experimentais

Para medir a eficiência do sistema híbrido de reconhecimento utilizamos cinco bases de dados: Dbham, db1, db2,

db3 e db4. A primeira (Dbham) é composta por 120 imagens do dedo indicador direito de 24 estudantes de nossa Universidade. Tais imagens foram capturadas pelo sensor Hamster¹. As demais bases de dados (Db1, Db2, Db3 e Db4 (subset B)) foram obtidas do Fingerprint Verification Competition (FVC 2006)². Estas bases são compostas por 120 imagens de 10 diferentes indivíduos, obtidas através de diferentes sensores. Em todas as bases listadas acima, nenhuma restrição de posição ou orientação foi imposta e as imagens variam em termos de qualidade. A Figura 2 apresenta exemplos de imagens das cinco bases em questão.



Figura 2. Imagens das bases usadas.

O sistema proposto opera com dois diferentes métodos de reconhecimento: SA e FC. Para verificar a eficiência do primeiro método quanto à combinação de informação local e global, cada imagem pertencente a uma determinada base de dados foi testada com todas as outras imagens restantes da mesma base. Com isso, um limiar é definido para permitir uma taxa de falsa aceitação (FAR - *False Acceptance Rate*) de 0%, garantindo que duas digitais de indivíduos diferentes nunca sejam aceitas. Os resultados apresentados na Tabela 1 mostram que a combinação dos dois escores melhora a taxa de aceitação em comparação ao uso de um ou outro escore isoladamente. Também é possível notar que o escore referente ao alinhamento das cristas produz resultados melhores do que aqueles originados pelo outro escore em imagens com reduzido número de minúcias ou com baixa qualidade (*e.g.* imagens das bases Db1 e Db3).

Experimentos com variação do FAR e Rank [1] foram executados para o FC e SA visando verificar a precisão dos

1 <http://www.secugen.com>

2 <http://bias.csr.unibo.it/fvc2006/>

	Dbham	Db1	Db2	Db3	Db4
M	27%	5%	51%	21%	30%
C	38%	36%	58%	27%	22%
M+C	70%	40%	92%	45%	45%

Tabela 1. Resultados do SA para minúcias (M), cristas (C) e combinação (M+C).

métodos. No RankN, uma dada amostra é comparada com todas as demais contidas na base de dados. Os escores originados destas comparações são ordenados e se a identidade estiver entre as N comparações com os menores escores (maior similaridade) então a amostra em questão foi identificada. A Tabela 2 apresenta os resultados do FC, e a Tabela 3 mostra os resultados do SA.

	Dbham	Db1	Db2	Db3	Db4
Rank1	82%	84%	99%	80%	85%
Rank5	93%	93%	99%	88%	96%
0%FAR	63%	35%	87%	36%	36%
1%FAR	70%	40%	89%	41%	44%
2%FAR	73%	44%	90%	47%	50%

Tabela 2. Resultados em função do Rank e do FAR para o FC.

	Dbham	Db1	Db2	Db3	Db4
Rank1	91%	91%	100%	85%	92%
Rank5	93%	98%	100%	90%	92%
0%FAR	70%	40%	92%	45%	45%
1%FAR	80%	54%	97%	60%	70%
2%FAR	82%	60%	98%	63%	75%

Tabela 3. Resultados em função do Rank e do FAR para o SA.

Como pode ser verificado, o alinhamento (SA) apresenta os melhores resultados em ambos os experimentos. Entretanto, esta abordagem não é aplicável a sistemas reais de identificação. Isto se deve ao fato de que o SA é capaz de executar somente cerca de cinco comparações por segundo em um Pentium D3.4GHz. Por outro lado, o FC pode executar eficientemente cerca de 15.000 comparações por segundo e apresenta resultados em função do Rank equivalentes aos do SA. Com base nisso, optou-se por usar o FC no modo de identificação e o SA no modo de verificação.

Considerando resultados de Rank1 para o FC e resultados de 0%FAR para o SA, obteve-se 79%, 72%, 97%, 77% e 73% como taxa de reconhecimento para as bases DbHam, Db1, Db2, Db3 e Db4 respectivamente. Usando Rank5 ao invés de Rank1 para o FC, elevou-se as taxas de aceitação para 88%, 78%, 98%, 89% e 92% para as bases DbHam, Db1, Db2, Db3 e Db4, respectivamente.

A performance do sistema híbrido depende da qualidade das imagens, da detecção do ponto de referência e das propriedades geométricas das digitais. Neste último caso, o SA pode confundir uma digital com outra que possua forma similar (arco, tenda, loop, etc). Quanto ao segundo caso, o sistema falha quando o ponto de referência não for precisamente detectado ou estiver muito próximo às extremidades da digital. Isto acontece porque o FC setoriza uma região de interesse em torno do ponto de referência que também é utilizado no pré-alinhamento do SA.

Quanto ao tempo de resposta, importante em sistemas reais de autenticação, a Tabela 4 mostra o tempo médio em segundos para cada módulo: (1) geração dos templates e processamento das imagens (PI-GT); (2) identificação (tempo médio para 119 comparações); e (3) verificação (tempo médio para 5 comparações). Como pode ser visto, o pior tempo de resposta foi de 2.1 segundos, qualificando o sistema proposto quanto à aplicações em tempo real [4].

	PI-GT	Ident.	Verif.	Total
DbHam	0.3	0.00595	1.0	1.30595
Db1	0.2	0.00595	0.4	0.60595
Db2	0.6	0.00595	1.5	2.10595
Db3	0.5	0.00595	1.4	1.90595
Db4	0.4	0.00595	1.1	1.50595

Tabela 4. Tempo de resposta do sistema.

5. Considerações Finais

Neste trabalho apresentamos um sistema automático de autenticação de impressões digitais atuante tanto no modo de identificação quanto no modo de verificação. Este sistema foi desenvolvido objetivando sua aplicação em sistemas de controle de acesso e em sistemas de login em nossa Universidade. O sistema também se apresenta como uma solução alternativa de baixo custo em relação às existentes no mercado, pois foi desenvolvido em software livre e estará disponível em breve à comunidade interessada.

Cabe reiterar que não há sistemas biométricos capazes de produzir respostas ótimas, variando de acordo com alguns critérios tais como a aceitação e a performance. De acordo com tais critérios, foi desenvolvido este sistema híbrido,

solução eficiente para o reconhecimento de impressões digitais. Este sistema utiliza dois métodos visando obter uma melhor relação entre taxa de acerto, tempo de resposta e taxa de aceitação. Estas duas técnicas utilizam tanto características globais quanto locais de uma imagem de impressão digital para verificar a identidade do indivíduo.

Os resultados experimentais mostraram que nossa abordagem híbrida é capaz de melhorar a taxa de reconhecimento quando comparada aos resultados dos dois métodos isoladamente. O algoritmo de segmentação pode detectar com eficiência regiões corrompidas na digital e removê-las das etapas seguintes do processo. Esta é uma propriedade muito importante uma vez que tais regiões afetam negativamente a extração das características necessárias ao reconhecimento, gerando informações falsas. A performance do sistema em relação ao tempo de resposta total, também é uma grande vantagem, uma vez que o tempo de identificação pode ser relativamente pequeno em bases de dados contendo até 20.000 amostras. Isto qualifica o sistema proposto quanto a sua utilização em aplicações em tempo real empregadas em bases de tamanho médio.

Referências

- [1] R. M. Bolle, J. H. Connell, S. Pankanti, N. K. Ratha, and A. W. Senior. *Guide to Biometrics*. Springer Professional Computing, Springer, New York, 2003.
- [2] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, 1992.
- [3] C. M. Holt, A. Stewart, M. Clint, and R. H. Perrot. An improved parallel thinning algorithm. *Communications of the ACM*, 30:156–160, 1987.
- [4] L. Hong, Y. Wan, and A. K. Jain. Fingerprint image enhancement: Algorithm and performance evaluation. *IEEE PAMI*, 20(8):777–789, 1998.
- [5] A. K. Jain, L. Hong, and R. Bolle. On-line fingerprint verification. *IEEE PAMI*, 19(4):302–314, 1997.
- [6] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti. Filterbank-based fingerprint matching. *IEEE Trans. on Image Processing*, 9(5):846–859, 2000.
- [7] X. Jiang and W. Y. Yau. Fingerprint minutiae matching based on the local and global structures. *IEEE ICPR*, 2:1042–1045, 2000.
- [8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [9] R. P. Lemes, M. P. Segundo, O. R. P. Bellon, and L. Silva. Registro de imagens no reconhecimento de impressões digitais. *V WUW - XX SIBGRAPI*, 2007.
- [10] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer, New York, 2003.
- [11] A. Ranade and A. Rosenfeld. Point pattern matching by relaxation. *IEEE ICPR*, 12(2):269–275, 1993.
- [12] A. Ross, A. Jain, and J. Reisman. A hybrid fingerprint matcher. *IEEE ICPR*, 36(7):1661–1673, 2003.
- [13] J. Ton and A. K. Jain. Registering landsat images by point matching. *IEEE GRS*, 27:642–651, 1989.

Fecho Convexo e o paradigma *Rotating Calipers*

Renato de J. Manzoni

Aluno de Graduação

Bacharelado em Ciência da Computação

FCT/UNESP - Campus de P. Prudente

rjmanzoni@gmail.com

Marco A. Piteri (orientador)

Faculdade de Ciências e Tecnologia - UNESP

Depto de Matemática, Estatística e Computação

Rua Roberto Simonsen, 305, Caixa Postal 467

Presidente Prudente, SP, Brasil

piteri@fct.unesp.br

Resumo

O desenvolvimento e a busca por eficientes abordagens algorítmicas voltadas para a obtenção da estrutura topológica e combinatória do fecho convexo de um conjunto de n pontos em \mathbb{R}^n é um dos problemas mais estudados na área de Geometria Computacional e historicamente foi um dos primeiros problemas geométricos a ser completamente analisado no plano. Uma outra importante característica associada ao fecho convexo é que outros problemas geométricos são facilmente resolvidos a partir de sua existência. Este trabalho explora o uso do paradigma de rotating calipers associado a estrutura do fecho convexo de um conjunto de n pontos no plano e mostra como estas duas abordagens podem solucionar vários outros problemas de natureza geométrica.

1. Introdução

O problema de se encontrar a estrutura do fecho convexo de um conjunto de n pontos no espaço \mathbb{R}^n possui inúmeros interesses teóricos e práticos. Entretanto, as aplicações mais imediatas estão associadas às dimensões 2 e 3, como por exemplo: detecção de colisão [1] em movimentos de robôs, cálculo da área/volume mínimo necessário para envolver um objeto 2D/3D e análise de formas [2], para citar algumas. De modo a limitar a abrangência do tema, no contexto desse trabalho vamos restringir a discussão ao espaço euclídeo de dimensão 2.

Formalmente, o *fecho convexo* de um conjunto finito de pontos em \mathbb{R}^2 , vamos denotar por $\text{Conv}(S)$, é o menor conjunto convexo que contém todos os pontos de S . Existem inúmeras e eficientes abordagens algorítmicas para se encontrar a solução desse problema, de forma exata ou aproximada [2] e já implementamos várias delas num projeto de iniciação científica suportado pela FAPESP. Contudo, o objetivo central desse trabalho é ilustrar como o paradigma

de *rotating calipers* pode ser combinado com a estrutura do fecho convexo para solucionar problemas de natureza geométrica envolvendo conjunto de pontos no plano, como por exemplo, o seu diâmetro [3], ou, as cascas convexas desse conjunto, a partir das quais é possível obter o nível de profundidade de um ponto no respectivo conjunto, ou ainda, uma triangulação dos pontos dados [5]. Além disso, vamos explorar como a idéia de *rotating calipers* pode ser usada para encontrar as tangentes inferior e superior, que é um passo construtivo fundamental para a obtenção da estrutura do fecho convexo quando se usa o princípio de *divide and conquer* [4]. Vale salientar ainda que a estrutura topológica do fecho convexo está diretamente relacionada a outras duas importantes subdivisões planares associadas a um conjunto de pontos arbitrários, mais especificamente, a *Triangulação de Delaunay* e o seu dual, que é o *Diagrama de Voronoi* [2].

2. Diâmetro do Fecho Convexo

Dado um conjunto de pontos $S = \{p_1, p_2, p_3, \dots, p_n\}$, $S \subseteq \mathbb{R}^2$, o *diâmetro* de S ($\text{diam}(S)$) é dado pela medida d_{\max} que maximiza a distância Euclideana entre quaisquer pares de pontos em S . Esta definição pode ser formalizada pela relação (1), dada abaixo.

$$d_{\max} = \max\{d(p_i, p_j), \forall p_i, p_j \in S, 1 \leq i < j \leq n\} \quad (1)$$

É fácil verificar que o par de pontos mais distantes de S está necessariamente em seu fecho convexo. Em outras palavras, o diâmetro de S é igual ao diâmetro dos pontos pertencentes ao $\text{Conv}(S)$. Existem algumas alternativas para se obter o $\text{diam}(S)$ a partir da existência de $\text{Conv}(S)$, mas no contexto desse trabalho vamos ilustrar a solução baseada no princípio de *rotating calipers* desenvolvida por Shamos [3].

Um par de pontos (p, q) no fecho convexo é denominado *antipodal* se existirem dois hiperplanos paralelos l_1 e

l_2 passando por p e q respectivamente, tal que, todos os outros pontos do fecho convexo estão entre l_1 e l_2 . A Figura 1 ilustra diagramaticamente esse conceito. A partir dessa mesma figura é possível observar que o diâmetro de um conjunto arbitrário S é dado por um par de pontos antipodal. Além disso, todos os pares de pontos antipodais se localizam em $Conv(S)$. Logo, para determinar o diâmetro de um conjunto S qualquer, é suficiente encontrar todos os pares de pontos antipodais e verificar aquele que maximiza a distância entre eles.

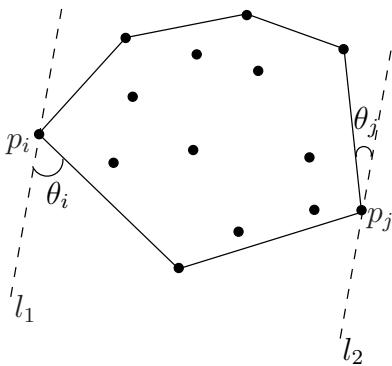


Figura 1. O par de pontos (p_i, p_j) é antipodal.

Para entender o processo de como todos os pares de pontos antipodais podem ser identificados, vamos considerar o polígono convexo ($Conv(S)$) da Figura 2, cujos pontos estão orientados no sentido anti-horário e determinar aqueles que são antipodais com relação ao ponto p_i .

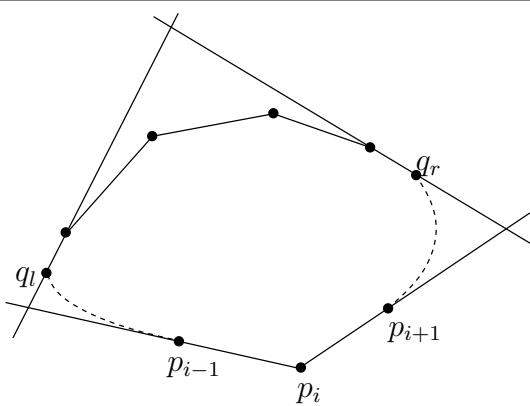


Figura 2. Encontrando todos os pontos $q \in Conv(S)$, tal que (p_i, q) seja antipodal.

Percorrendo os pontos no sentido anti-horário, encontra-

mos o ponto q_r , tal que, a distância entre a reta suporte passando pelo segmento $\overline{p_{i-1}p_i}$ a q_r , seja a maior possível. Semelhantemente, a partir de p_i e, “caminhando” no sentido horário, encontramos o ponto q_l , tal que, a distância entre a reta suporte passando por $\overline{p_ip_{i+1}}$ a q_l seja maximizada. Logo, a cadeia de pontos entre q_r e q_l (sentido anti-horário), incluindo seus extremos, define o conjunto $C(p_i)$ de pontos que formam pares antipodais com relação ao ponto p_i . Essa idéia é repetida para os demais pontos, o que permite identificar todos os pares de pontos antipodais. Devemos observar que essa abordagem não exige em nenhum momento a necessidade do cálculo de ângulos, aumentando a eficiência do processo computacional envolvido.

Para explorar e entender um pouco mais a idéia intuitiva do uso do paradigma *rotating calipers* aplicado na obtenção dos pares de pontos antipodais, vamos considerar a Figura 1, onde pode ser observado que o par de pontos (p_i, p_j) são antipodais. Essa conclusão pode ser corroborada por meio da observação das retas tangentes l_1 e l_2 , que passam por p_i e p_j , respectivamente. Vamos considerar ainda que os pontos pertencentes ao polígono convexo estejam orientados no sentido anti-horário.

Para encontrar o próximo ponto antipodal em relação a p_i , seja θ_i o ângulo formado pela reta l_1 com a aresta $\overline{p_ip_{i+1}}$ e θ_j , o ângulo entre a reta l_2 com a aresta $\overline{p_jp_{j+1}}$, ver Figura 1. Se $\theta_j < \theta_i$, então rotacionamos ambas as retas de um ângulo igual a θ_j , encontrando o ponto p_{j+1} que é antipodal com relação a p_i . Este procedimento continua até voltarmos à posição original.

3. Triângulação de Pontos a partir de Cascas Convexas

As *cascas convexas* de um conjunto de pontos S pode ser obtida a partir da aplicação sistemática do algoritmo de fecho convexo. Inicialmente, o algoritmo é aplicado ao conjunto original de pontos S , dando origem a primeira camada convexa. Em seguida, o conjunto de pontos pertencentes a $Conv(S)$ é retirado de S , e, sobre o conjunto resultante, executa-se novamente o algoritmo de fecho convexo, o que resulta na segunda camada. Esse procedimento é repetido até que não haja mais pontos para serem trabalhados. A Figura 3 ilustra um exemplo de cascas convexas obtidas a partir de um conjunto de pontos arbitrário.

Encontrar as cascas convexas de um conjunto de pontos S , permite que possamos obter a *profundidade convexa* de um ponto p_i qualquer neste conjunto, que é definida como o número de cascas convexas que devem ser removidas de S , até que p_i seja um ponto extremo. Assim, os pontos com profundidade convexa 0 são aqueles que pertencem a $Conv(S)$. Por outro lado, os pontos com profundidade convexa 1 são todos os pontos pertencentes a $Conv(S - Conv(S))$. Em outras palavras, os pontos do

fecho convexo obtido a partir da remoção do conjunto S , dos pontos $Conv(S)$. Entretanto, nosso maior interesse na determinação das cascas convexas de um conjunto de pontos $S \subseteq \mathbb{R}^2$, é obter uma triangulação desses pontos. Para esse propósito é necessário definirmos o conceito de *anel*, que é a região planar compreendida entre duas cascas convexas consecutivas, conforme pode ser observado na Figura 4.

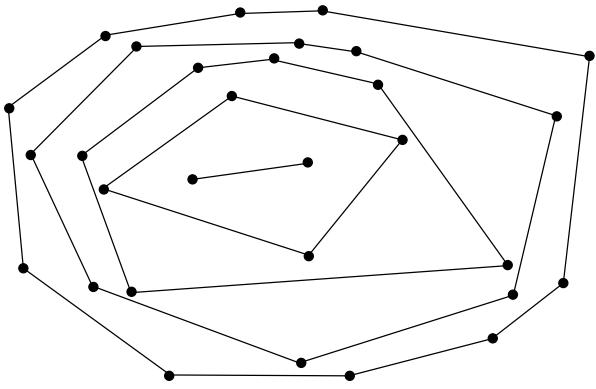


Figura 3. Exemplo de cascas convexas associadas a um conjunto de pontos no plano.

O algoritmo de triangulação de pontos no plano a partir das cascas convexas e que utiliza a idéia de *rotating calipers* é devido a Toussaint [5] e possui complexidade de tempo linear.

Sejam P e Q dois conjuntos de pontos orientados no sentido anti-horário e associados, respectivamente, a duas cascas convexas simultâneas em S . Todas as arestas pertencentes as cascas convexas são também arestas da triangulação de S , portanto, é suficiente obter as arestas localizadas no anel formado por P e Q . O passo inicial é computar os pontos de abscissa mínima p_i e q_j nos conjuntos P e Q , respectivamente. A aresta ligando ambos pertence a triangulação. A Figura 4 ilustra diagramaticamente a descrição acima. Em seguida, contrói-se as retas de suportes verticais l_p e l_q (*calipers*) passando pelos pontos p_i e q_j , respectivamente.

O próximo passo é rotacionar (*rotating*) essas retas no sentido anti-horário até que uma delas coincida com uma das arestas de P e Q , encontrando um novo vértice (p_{i+1} ou q_{j+1}), que por sua vez dará origem a uma nova aresta da triangulação. Se este vértice pertencer a P , então, a aresta resultante será (p_{i+1}, q) , caso contrário, a nova aresta será definida por (p, q_{j+1}) . No caso de ambas as retas suportes coincidirem com as arestas (arestas paralelas), é indiferente escolher uma ou outra. Este procedimento é repetido até alcançar novamente os pontos de origem p_i e q_j . É importante observar que a triangulação obtida a partir dessa

técnica pode ser facilmente convertida numa *triangulação de Delaunay* por meio da aplicação do princípio de *edge-flip* [2].

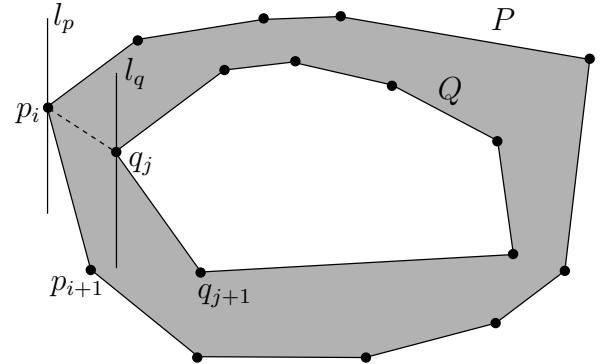


Figura 4. Triangulação de pontos a partir do princípio de *rotating calipers* aplicado a região sombreada denominada *anel*.

4. Combinando Fechos Convexos

Uma outra aplicação do princípio de *rotating calipers* é na obtenção das tangentes inferior e superior necessárias para obter a união de dois conjuntos convexos arbitrários e disjuntos, mantendo-a convexa. Esse passo construtivo é utilizado para encontrar $Conv(S)$ a partir da abordagem *dividir-para-conquistar* [2].

Para combinar (unir) dois fechos convexos intermediários P e Q , devemos encontrar os pontos $p_t, p_k \in P$ e $q_r, q_s \in Q$, tal que, os segmentos $\overline{p_t q_r}$ e $\overline{p_k q_s}$, sejam tangentes aos conjuntos P e Q e os pontos de ambos os conjuntos estejam compreendidos entre as tangentes. Em outras palavras, se olharmos para cada tangente individualmente, os pontos de $P \cup Q$ estão de um mesmo lado.

Toussaint [4] propôs aplicar a noção de *rotating calipers* para identificar essas retas tangentes. Vale salientar que esse mesmo passo algorítmico é usado para solucionar vários outros problemas na área de Geometria Computacional.

Para resolver o problema de encontrar os pares de pontos que dão origem as retas tangentes que resolvam o problema proposto, é necessário conhecer o conceito de pares de pontos co-podais. Dados dois polígonos P e Q , um par de pontos (p, q) ($p \in P$ e $q \in Q$) é denominado co-podal entre P e Q , se os respectivos polígonos admitem retas de suporte passando por p e q , respectivamente, que sejam paralelas. A Figura 5 ilustra um par de pontos co-podais.

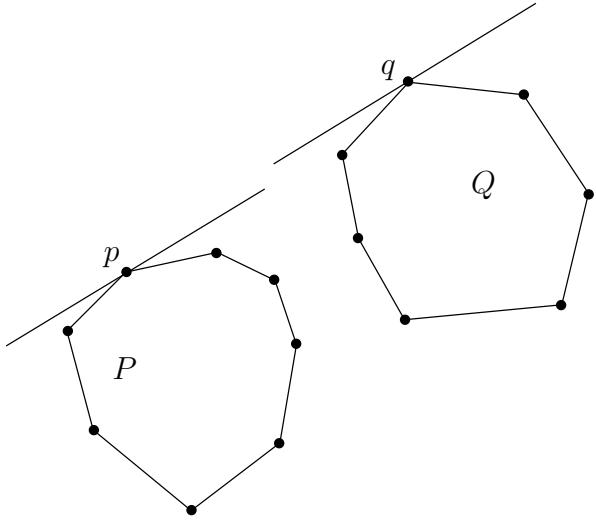


Figura 5. Exemplo de um par de pontos (p, q) co-podal.

Dois vértices $p_i \in P$ e $q_j \in Q$ pertencem a uma das tangentes procurada, se e somente se, eles satisfazem as duas condições abaixo:

condição 1 os vértices p_i e q_j formam um par *co-podal*;
condição 2 os vértices p_{i-1} , p_{i+1} , q_{j-1} e q_{j+1} , estão todos de um mesmo lado em relação ao segmento de reta $\overline{p_i q_j}$.

A Figura 6 ilustra o exemplo do par de pontos p_i e q_j que é co-podal, porém, não satisfaz a **condição 2**. Assim, não pode ser uma tangente que define uma aresta de $P \cup Q$. Observe ainda que que na Figura 6, os pontos p_{i-1} e q_{j-1} referenciados na **condição 2** estão representados por p_k e q_r , respectivamente.

A partir desses conceitos é fácil elaborar um algoritmo para encontrar as duas tangentes de interesse, já que os pares co-podais são gerados durante o *rotating calipers*. Logo, é suficiente verificar se os vértices adjacentes aos pares de vértices co-podais, situam-se de um mesmo lado em relação a reta suporte passando pelos vértices co-podais. Caso isto seja satisfeito, os vértices co-podais são os pontos que definem as tangentes procuradas. No caso do exemplo dado na Figura 6, os pares de pontos co-podais são formados por (p_t, q_r) (tangente superior) e (p_k, q_s) (tangente inferior).

5. Conclusão e Trabalhos Futuros

Este trabalho investiga a aplicação do paradigma de *rotating calipers* combinado com a estrutura do fecho convexo, objetivando simplificar a solução de problemas envolvendo conjunto de pontos no plano. Além disso, ilustra como esse

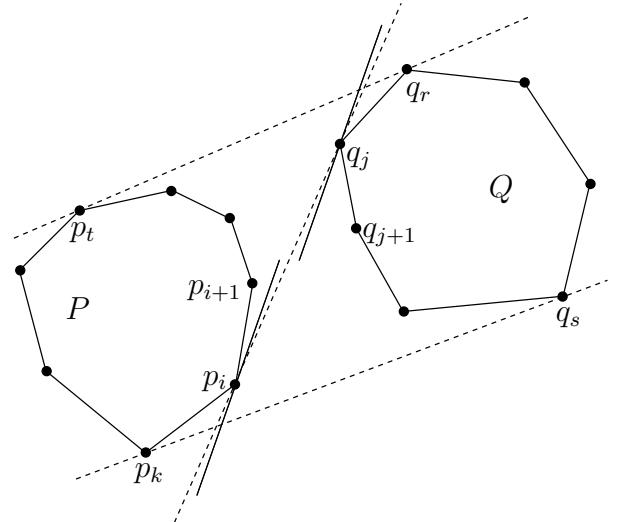


Figura 6. Obtenção de retas tangentes usando o conceito de par de pontos co-podal.

método pode ser usado de forma eficiente para a obtenção de tangentes específicas associadas a dois polígonos convexos. Esse passo construtivo é muito comum em problemas geométricos cuja solução envolve o paradigma de *dividir-para-conquistar*. Na continuidade do trabalho, pretendemos verificar possíveis aplicações do problema do cálculo do diâmetro de um conjunto de pontos, fazendo uso de outras métricas.

Agradecimentos

O autor e o co-autor agradecem a FAPESP pela bolsa de Iniciação Científica concedida, bem como os pareceristas do artigo pelas sugestões apresentadas, que em muito contribuíram para a qualidade final do trabalho.

Referências

- [1] T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM*, 22(10):560–570, 1979.
- [2] F. P. Preparata and M. I. Shamos. *Computational geometry: An Introduction*. Springer-Verlag, 1988.
- [3] M. Shamos. *Computational geometry*. PhD thesis, Yale University, 1978.
- [4] G. Toussaint. Solving geometric problems with the rotating calipers. Proc. MELECON, 1983.
- [5] G. T. Toussaint. A linear-time algorithm for solving the strong hidden-line problem in a simple polygon. *Pattern Recogn. Lett.*, 4(6):449–451, 1986.

Rastreamento facial em vídeo com aplicação em monitoramento e Segurança

Alessandro G. F. Fior, Maurício P. Segundo, Luciano Silva, Olga R. P. Bellon*

Grupo IMAGO de Pesquisa – Universidade Federal do Paraná

Caixa Postal 19092 – CEP 81531-980 – Curitiba/PR – Brasil

{agff05, mauricio, luciano, olga}@inf.ufpr.br

Resumo

Neste trabalho, apresentamos uma ferramenta para rastreamento de faces em seqüências de vídeo que é componente de um sistema de monitoramento e Segurança baseado em informações multibiométricas. A ferramenta tem como diferencial uma forte integração com a etapa de detecção de faces, anterior ao rastreamento. Inicialmente, a aplicação busca por faces no vídeo combinando detecção de movimento, filtragem de pele, e um classificador baseado em características de Haar para diferenciar imagens faciais e não-faciais. Este mesmo classificador é aplicado para o rastreamento, uma vez que as características de Haar mantêm informações do padrão das faces. Nenhuma informação adicional é necessária para o rastreamento, apenas as características já calculadas durante a etapa de detecção. Os experimentos foram realizados em uma base própria de seqüências de vídeo adquiridas em um ambiente interno. Com a abordagem proposta, a porcentagem de faces com a posição corretamente estimada foi de 94,18%.

1. Introdução

Os sistemas de reconhecimento biométrico através de faces em vídeo [4, 5, 9] compreendem, de modo geral, três fases: (1) detecção de faces [10, 11], (2) rastreamento das faces ao longo do vídeo [12] e (3) reconhecimento da identidade das faces [2]. O rastreamento tem como função garantir a continuidade das faces em uma seqüência de vídeo, assegurando que uma mesma face encontrada em vários quadros do vídeo pertence à mesma pessoa. Para isso, o rastreador estima a localização de uma face mesmo que ela não seja encontrada pelo detector em um ou mais quadros.

Nesse trabalho é proposto um método de rastreamento facial cujo diferencial é a integração com a etapa de detecção. O método é baseado em características de Haar [10], amplamente utilizadas para a detecção facial. O mesmo conjunto de características aplicado na

detecção é utilizado no rastreamento, que é mais robusto a variações ao longo das seqüências de vídeo (e.g. variações de iluminação e expressão) do que a detecção.

Os métodos de rastreamento de alvos disponíveis na literatura utilizam diferentes tipos de informação, como por exemplo, histograma da intensidade dos pixels do alvo [3] ou a covariância da intensidade e/ou gradiente dos pixels do alvo [8]. Ao utilizar algumas destas abordagens, é necessário calcular características específicas para o rastreamento, enquanto a abordagem proposta utiliza as mesmas características já calculadas para a detecção de faces.

O objetivo deste trabalho é monitorar pessoas que circulam em um determinado ambiente. Neste contexto, para a realização dos experimentos, foi criada uma base própria contendo seqüências de vídeo. A criação dessa base foi necessária pois não havia nenhuma base pública disponível com as características desejadas: pessoas andando em direção à câmera, com variações de iluminação, expressão e dimensão das faces.

O método de rastreamento proposto é apresentado na Seção 2 deste artigo. Os resultados experimentais são mostrados na Seção 3, seguidos de algumas considerações finais e referências utilizadas.

2. Rastreamento facial em vídeo

O método de rastreamento proposto neste artigo é baseado em uma cascata de classificadores formados por características de Haar. Estes classificadores são gerados inicialmente para realizar a detecção de faces [10], mas fornecem um vetor característico que permite diferenciar a região detectada das demais regiões da imagem.

2.1. Características de Haar para detecção facial

Neste trabalho, 7 tipos de características de Haar foram utilizadas [6]. Seus valores representam a diferença do somatório de pixels em diferentes áreas de uma máscara, que varia conforme o seu tipo. Estas características são aplicadas com diferentes dimensões e posições na imagem.

* Os autores gostariam de agradecer ao CNPq, CAPES e FINEP pelo suporte financeiro.

Quando estas características são aplicadas em padrões similares (*e.g.* faces), estes valores tendem a ser próximos. Logo, eles podem ser utilizados para identificar se uma imagem pode ou não ser de um determinado padrão. Embora uma única característica de Haar não seja suficiente para diferenciar grandes quantidades de imagens de faces e não-faces, classificadores mais eficientes podem ser obtidos combinando várias características.

A partir de uma base contendo imagens de faces e não-faces, o algoritmo *AdaBoost* [10] é aplicado para encontrar as características que melhor diferenciam imagens de faces e não-faces e distribuir estas características em classificadores. Os primeiros classificadores gerados possuem menos características que os seguintes, e assim consecutivamente. A Fig. 1 mostra dois exemplos de classificadores com diferentes quantidades de características de Haar.

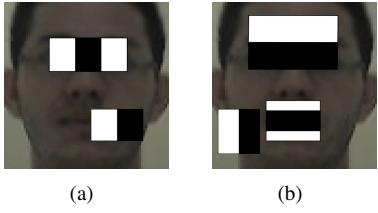


Figura 1. Exemplo de classificadores contendo (a) 2 e (b) 3 características de Haar.

Para a detecção, uma subjanela percorre a imagem selecionando como faces apenas as regiões aceitas por todos os classificadores gerados, que são aplicados em ordem crescente de complexidade para melhorar o desempenho do método. Outra forma de otimizar a detecção consiste em realizar a busca apenas em regiões da imagem que apresentem movimento [1] e cor de pele [7].

2.2. Rastreamento facial

Quando uma face f é detectada, esta pode ser representada como um vetor de L classificadores, e cada classificador como um vetor de valores das características de Haar $C_i^f = [h_f(i, 1), h_f(i, 2), \dots, h_f(i, M_i)]$, onde C_i^f é o i -ésimo classificador da face f , contendo M_i características de Haar, e $h_f(i, j)$ é o valor da j -ésima característica deste classificador.

Para melhorar o desempenho do rastreador, apenas um subconjunto dos classificadores é utilizado. Segundo Yao e Li [12], quando um detector é construído em cascata, os primeiros classificadores utilizam características grosseiras, enquanto os últimos classificadores representam melhor a face. Por este motivo, neste trabalho apenas os L' últimos classificadores são usados para o rastreamento.

Esta representação é gerada para cada subjanela em uma vizinhança em torno da face detectada. A distância E entre uma face f e uma subjanela g é dada pela soma das distâncias Euclidianas entre os classificadores em f e seus respectivos classificadores em g , como mostrado na Eq. 1:

$$E(f, g) = \sum_{i=L-L'+1}^L \sqrt{\sum_{j=1}^{M_i} (h_f(i, j) - h_g(i, j))^2} \quad (1)$$

A posição e o tamanho da face rastreada correspondem aos valores da subjanela g do quadro atual com a menor distância da face f . A representação da face é atualizada para o próximo quadro segundo a Eq. 2:

$$f_{t+1} = \alpha g + (1 - \alpha) f_t \quad (2)$$

onde α é a taxa de aprendizado do rastreador, f_t é a representação atual da face, e f_{t+1} é a nova representação.

Entretanto, estas atualizações serão necessárias somente se a face rastreada não for detectada no quadro atual. Se a face for detectada, o rastreamento é utilizado apenas para garantir que esta pertence à seqüência do mesmo indivíduo. Neste caso, a posição e o tamanho da face a ser rastreada são atualizados com as informações da face detectada h , e a representação é atualizada segundo a Eq. 3:

$$f_{t+1} = \beta h + (1 - \beta) f_t \quad (3)$$

onde β é a taxa de aprendizado usada para faces detectadas.

3. Resultados experimentais

Para a realização dos experimentos, foi criada uma base de validação contendo 20 seqüências de vídeo de 5 indivíduos. Cada seqüência possui em média 127 quadros, e foi adquirida por uma câmera de segurança AXIS 207MW a 18Hz com resolução de 1280×720 pixels. A Fig. 2 mostra um dos quadros de uma seqüência da base.



Figura 2. Exemplo de um quadro de uma seqüência de vídeo da base.

Estas seqüências contêm um indivíduo andando em um corredor, e as faces apresentam variações como oclusões parciais e diferenças de iluminação, escala, pose e expressões. Algumas destas variações são mostradas na Fig. 3.

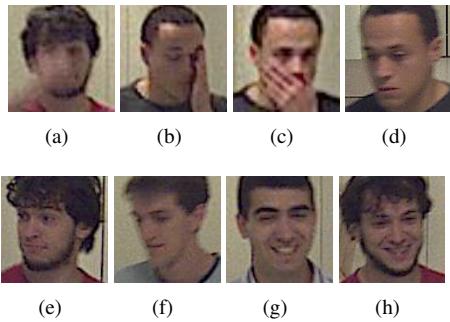


Figura 3. Imagens apresentando (a)-(c) oclusões parciais, (d)-(f) variações de pose, e (g)-(h) expressões faciais.

3.1. Detecção facial

Para treinar o detector, foi utilizada uma base de treino com 4483 imagens de faces e 74065 imagens de não faces. As imagens de treino são diferentes das contidas na base de validação. A cascata resultante contém 264 características de Haar distribuídas em 28 classificadores.

Em todas as seqüências da base de validação as faces são detectadas. Os dois primeiros classificadores da cascata rejeitam em torno de 82% das imagens de não-faces. Em média, apenas 4 características foram aplicadas por subjanela devido à organização em cascata dos classificadores.

3.2. Rastreamento facial

Para esta etapa dos experimentos, cada face foi marcada manualmente nas seqüências para serem utilizadas como referência para o rastreamento. A partir destas referências, foi possível medir o erro em pixels do deslocamento nos eixos x e y, e da escala para cada seqüência. Aplicando um limiar de tolerância neste erro, é possível determinar se a face foi rastreada corretamente. Esse limiar é necessário porque podem haver pequenas diferenças entre a marcação manual e a posição determinada pelo rastreador. Nos resultados apresentados nessa Seção, utilizamos um limiar de 15% do tamanho da face. Ou seja, se a distância entre os centros das faces rastreada e de referência ou a diferença entre seus tamanhos forem maiores do que 15% do tamanho da face de referência, então o rastreamento é considerado incorreto.

As seqüências de vídeo foram utilizadas para determinar os parâmetros α e β . Para estimar a taxa de aprendizado α do rastreador, o valor de β foi definido como 0 e diferentes valores de α foram testados. A Fig. 4 mostra a porcentagem média de acerto do rastreador ao longo do tempo para alguns destes valores testados. O tempo 0 representa a primeira face rastreada, e o tempo 1 a última face da seqüência. As outras faces foram distribuídas ao longo deste intervalo.

Como pode ser observado, valores mais altos para α apresentam um melhor comportamento ao longo do tempo.

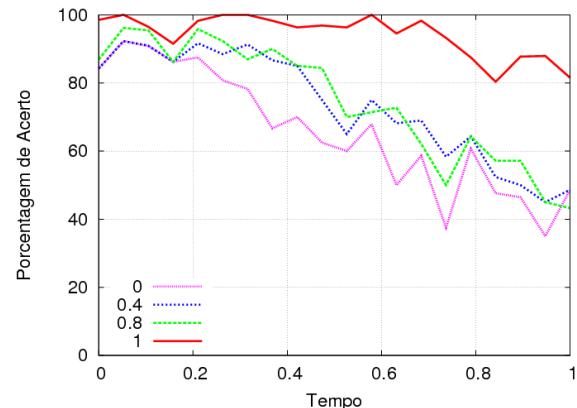


Figura 4. Erro médio do rastreador para diferentes valores de α .

O desempenho do rastreamento diminui no final das seqüências devido à baixa qualidade das imagens. A qualidade diminui por dois motivos: (1) borrões causados por movimentação acentuada, e (2) câmera com foco fixo, que deixa os objetos próximos a ela desfocados. A Fig. 5 mostra alguns exemplos destas imagens.



Figura 5. Exemplos de quadros do final das seqüências com baixa qualidade.

Para estimar a taxa de aprendizado β para faces detectadas, o valor de α foi definido como 0. Foram testados para β os mesmos valores testados para α . A Fig. 6 mostra a porcentagem média de acerto ao longo do tempo para alguns destes valores, e, assim como para o parâmetro α , valores mais altos para β apresentam um desempenho melhor.

Em nossos experimentos, observamos que os valores das taxas de aprendizado estão relacionados à velocidade de captura da câmera. Quanto maior a quantidade de quadros por segundo, menor é a variação entre quadros consecutivos, diminuindo a necessidade de atualizar o modelo de rastreamento. Conseqüentemente, as taxas de aprendizado podem ser menores, e os modelos tornam-se mais robustos a variações no vídeo.

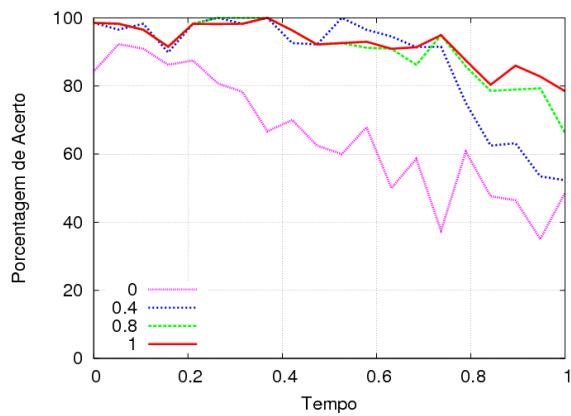


Figura 6. Erro médio do rastreador para diferentes valores de β .

A Fig. 7 mostra as faces detectadas e rastreadas de uma seqüência de vídeo da base. Como pode ser observado, muitos quadros não tiveram suas faces detectadas devido a variações de pose e oclusões, mas o rastreador foi capaz de estimar corretamente suas posições.



Figura 7. Faces detectadas (com borda preta) e rastreadas em uma das seqüências.

Computamos as porcentagens médias de acerto do rastreador para diferentes limiares de erro considerando a me-

lhor configuração obtida a partir dos experimentos anteriores ($\alpha = 1$, $\beta = 1$). Para o limiar de tolerância de 15% utilizado nos experimentos anteriores, localizamos corretamente a face em 94,18% dos quadros, para um limiar de 10%, esta porcentagem cai para 83,41%, e considerando um limiar de 40%, a porcentagem de acerto é de 98,82%.

4. Considerações finais

Neste trabalho apresentamos um método de rastreamento de faces que utiliza as mesmas características aplicadas para a detecção facial. O método foi capaz de relacionar as faces detectadas em todas as seqüências de vídeo, e estimar corretamente a posição da face em aproximadamente 94% dos quadros, considerando um limiar de erro de 15%. Os experimentos mostram que o rastreamento é mais robusto que a detecção quando as faces apresentam variações como oclusões parciais, ou diferenças de iluminação, pose e expressões. Esta abordagem está inserida em um sistema de monitoramento e Segurança multi-biométrico, responsável pela localização da informação a ser utilizada pelo reconhecimento facial.

Referências

- [1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *CVIU*, 73(3):428–440, 1999.
- [2] R. Chellappa, C. L. Wilson, and S. Sirohey. Human and machine recognition of faces: a survey. In *Proc. IEEE*, volume 83, pages 705–741, 1995.
- [3] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE PAMI*, 24(5):603–619, 2002.
- [4] A. Hadid and M. Pietikäinen. An experimental investigation about the integration of facial dynamics in video-based face recognition. *Electronic Letters on Computer Vision and Image Analysis*, 5(1):1–13, 2005.
- [5] V. Krueger and S. Zhou. Exemplar-based face recognition from video. In *Proc. FG*, pages 182–187, 2002.
- [6] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Proc. ICIP*, pages 900–903, 2002.
- [7] P. Peer, J. Kovac, and F. Solina. Human skin colour clustering for face detection. In *Proc. EUROCON*, 2003.
- [8] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on lie algebra. In *Proc. CVPR*, pages 728–735, 2006.
- [9] J. Steffens, E. Elagin, and H. Neven. PersonSpotter - fast and robust system for human detection, tracking and recognition. In *Proc. FG*, pages 516–521, 1998.
- [10] P. Viola and M. J. Jones. Robust real-time face detection. *Int. J. Computer Vision*, 57(2):137–154, 2004.
- [11] M. H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE PAMI*, 24(1):34–58, 2002.
- [12] Z. Yao and H. Li. Tracking a detected face with dynamic programming. In *Proc. CVPRW*, volume 5, pages 63–70, 2004.

Simulações de Movimentos Físicos em Duas Dimensões, Utilizando Framework XNA

Luiz Filipe F. Carneiro, Wietske Ineke Meyering (orientadora)
Faculdade Farias Brito
{luizfilipefc, wietske.meyering}@gmail.com

Resumo

Para que jogos 2D sejam produzidos com qualidade é necessário simular a realidade. Para isso conceitos de física mecânica e matemática devem ser aplicados no processo de codificação de um jogo, principalmente na emulação de circunstâncias reais, como uma bola batendo na borda da mesa de sinuca, óleo em uma pista de corrida como fator de redução de atrito ou um objeto em um espaço com gravidade desprezível. Os objetivos deste artigo são explanar os conceitos de física mecânica e matemática para a simulação de movimentos em duas dimensões de um determinado objeto e a aplicação destes em simuladores construídos em um ambiente na plataforma (ou Framework) XNA Game Studio.

1. Introdução

Para simular movimentos em 2D é necessário considerar conceitos físicos como inércia, força, ação e reação, movimento acelerado, movimento retardado, bem como o uso do plano cartesiano e vetorialização na matemática computacional para a interação de um objeto com o meio. Além disso, para lidar com a plataforma XNA é necessário ser proficiente com a linguagem de programação C#.

2. Aspectos Básicos da plataforma XNA

Ao se criar um novo projeto na plataforma XNA deve-se estar ciente de quais componentes foram criados e para que cada um deles serve. Logo após, verifica-se a função de cada um dos métodos que compõem a estrutura inicial do projeto.

Um novo projeto contém uma pasta chamada *Content* e dois arquivos, *Game1.cs* e *Program.cs*.

Game1.cs é o arquivo de código que carrega a lógica de um jogo e inicializa as variáveis de instância [4].

Program.cs é o arquivo de código que instancia um objeto de *Game1.cs* e executa o código do jogo [4].

Content guarda todo o conteúdo de mídia (Imagens, Fontes) que será utilizado durante o processo de desenvolvimento do jogo em XNA [4].

2.1. Sprites

Um *sprite* é uma imagem em duas dimensões que pode ser manipulada independentemente do cenário do jogo [4]. No XNA ela possui propriedades como largura, altura e velocidade.

Segundo Cawood [5], *sprite* é um conjunto de imagens guardadas em um único arquivo de imagens com o propósito de criar animação. Em jogos 2D *sprites* são freqüentemente usados em visão de terceira pessoa. Como componente, um *sprite* apresenta propriedades especiais de imagem como filtros *alpha* que torna possível o *alpha blending*, que é definido como o processo de combinação de cor de um objeto em primeiro plano translúcido, com a cor do fundo.

2.2. O “Game Loop”

O *game loop* é a repetição das rotinas cíclicas de um jogo, como entradas dos jogadores, cálculos que serão determinados pela inteligência artificial do jogo [4], detecções de colisão, etc. O *game loop* só encerra quando um critério de finalização do jogo é executado dentro de seu escopo [4]. Ele é constituído de dois métodos: *Update()*, que contém toda a lógica do jogo, e o método *Draw()*, que adiciona os objetos em suas possíveis novas posições. Estes métodos estão localizados na classe *Game1* [4].

2.3. Carregando o Conteúdo

Todo e qualquer conteúdo multimídia usado no jogo, além de ser alocado na pasta *Content*, deve ser instanciado na classe *Game1* no método *LoadContent()* para ser desenhado no método *Draw()* que faz parte do *game loop* [4]. O método *Initialize()* é utilizado para inicializar as variáveis de instância da classe *Game1* [4].

Para descarregar os objetos que não serão mais utilizados faz-se o uso do método *UnLoadContent()* [4].

3. Fundamentos Matemáticos aplicados à plataforma XNA

3.1. Plano Cartesiano

Um plano cartesiano é um espaço bidimensional constituído por dois eixos, onde um ponto é representado por um par ordenado na forma (x,y).

Todo plano cartesiano possui uma coordenada especial utilizada para definir o ponto de origem [1]. A plataforma XNA trata o canto superior esquerdo da tela do jogo como origem O (0,0). Nele, o eixo Y é orientado de cima para baixo e o eixo X da esquerda pra direita, como na definição de imagens em Computação Gráfica.

O campo visual do plano cartesiano computacional é delimitado pela largura e pela altura da janela do jogo fazendo com que este possua um ponto máximo, entretanto pode-se utilizar coordenadas fora do campo visual da janela do jogo.

3.2. Vetores em duas dimensões

Algumas grandezas, expressas por variáveis de instância no código de um jogo em XNA, devem possuir módulo, direção e sentido, portanto podemos definir força e velocidade como grandezas vetoriais que são diferentes de uma grandeza escalar como a massa de um objeto [2]. Um vetor em 2D (\vec{v}) pode ser representado por um segmento de reta que parte da origem e se encerra nas suas componentes (v_x, v_y).

3.2.1. Algumas operações básicas com vetores.

Negar um vetor [1] ou a Inversão da direção [3] significa mudar o sentido do vetor multiplicando as suas componentes cartesianas por menos um para obter um vetor com mesma direção e módulo, mas de sentido oposto. Um objeto com velocidade (\vec{v}) construído em uma simulação na plataforma XNA se desloca para direita em determinado momento e após a Negação de

seu vetor (\vec{v}) o objeto passa a se deslocar para esquerda.

O módulo de um vetor 2D (\vec{v}) tal que seus componentes sejam (v_x, v_y) é dado pela expressão a seguir:

$$|v| = \sqrt{v_x^2 + v_y^2}$$

Na prática vetores não são medidos pelo seu módulo, mas são expressos pela medida de suas coordenadas. Na plataforma XNA é possível obter esse resultado apenas utilizando um método *Length()* de um objeto da classe *Vector2* [5].

Multiplicar um vetor 2D por um escalar (i) é multiplicar o seu módulo por (i) ou de modo semelhante multiplicar todas as componentes desse vetor por (i) [1].

Segundo Dunn [1] existem situações mais fáceis de lidar quando se conhece o caminho do vetor. O sentido de um determinado vetor 2D é representada pela sua normalização.

Seja (\vec{a}) a normalização de um vetor 2D (\vec{v}), onde este é formado por componentes (\vec{v}_x, \vec{v}_y), é representada pela expressão abaixo:

$$\vec{a} = \frac{\vec{v}_x + \vec{v}_y}{|\sqrt{v_x^2 + v_y^2}|}$$

Contudo na Plataforma XNA o valor da normalização de um vetor pode ser obtido com o método *Normalize()* da classe *Vector2* [5].

A soma e a subtração de vetores consistem em realizar estas operações básicas nas componentes de dois ou mais vetores de duas dimensões.

4. Fundamentos da Física Mecânica Aplicados à Plataforma XNA

4.1. Tempo

O tempo é uma medida escalar fisicamente expressa em horas, minutos, segundos e suas frações.

As definições físicas do tempo ficam explícitas se o programador conseguir vincular o tempo ao *game loop*, imaginando que cada iteração é uma unidade de tempo.

4.2. Matéria

Para que um determinado *sprite* no plano cartesiano interaja com as bordas da janela ou com algum outro *sprite* ou obedeça a queda gravitacional é necessário atribuir à ele, massa e solidez. Dessa forma ele pode colidir com outros *sprites* e ser afetado por grande

parte dos fenômenos físicos importantes para a movimentação em duas dimensões.

Para tanto, deve-se delimitar o espaço ocupado por esses objetos e realizar testes de interseção. No XNA isso é feito usando o método *Intersects()* [4] de alguma das classes que implementam detecção de colisão. Existem dois tipos de testes a serem feitos em ambientes computacionais, os testes estáticos que retornam um valor booleano, verdadeiro se houver colisão ou falso caso contrário, e os testes de interseção dinâmicos que além de retornarem valores booleanos têm uma terceira componente temporal que indica quando os objetos colidem. O teste de colisão dinâmico é mais utilizado para colisão de raios e raramente é aplicado na simulação de movimento dos objetos [1].

Existem três tipos básicos de colisões estáticas na plataforma XNA que são amplamente utilizadas em desenvolvimento de jogos 3D e 2D são eles: *BoundingBox*, *BoundingSphere*, e colisão por pixel.

4.2.1. Colisões. Colisões por *BoundingBox* são utilizadas na edificação de um jogo em duas dimensões, quando o *sprite* em questão tem o formato retangular [4], para que não haja colisão entre as partes em *alpha* do *sprite*.

De forma similar à colisão por *BoundingBox*, as colisões por *BoundingSphere* [5] em duas dimensões, apenas podem ser realizadas em *sprites* de forma circular. Para colisões por *BoundingSphere* é necessário ter informações do centro e do raio da circunferência.

A colisão por pixel de simulações em 2D é a mais eficiente, complexa e custosa computacionalmente de se edificar [4]. A sua utilização se dá por meio de jogos que empregam o conceito de *sprites* animados com formatos complexos como um *sprite* em formato humanóide. O método desta colisão consiste em guardar a informação dos pixels da matriz que formam a imagem em um array e comparar se há colisão entre aqueles pixels que possuem informações *alpha* diferente de zero.

4.3. Leis de Newton Aplicadas à Simulação computacional

4.3.1. Inércia. Na iminência de uma força, um corpo tende a permanecer imóvel devido a inércia, que é uma propriedade da matéria. Um corpo permanece inerte em dois casos: quando se encontra em repouso ou em movimento uniforme (um exemplo disso é o movimento no vácuo), quando a força resultante atuando sobre o objeto é igual a zero, logo o corpo não sofre aceleração [2].

4.3.2. Força. Em XNA outro aspecto que pode ser abordado através do *game loop* é a força. A aplicação da simulação de movimento funciona de forma que se existe uma força resultante atuando sobre um determinado corpo na mesma direção da velocidade, a aceleração pode aumentar ou diminuir, dependendo do sentido da velocidade deste corpo (movimento acelerado, movimento retardado) em relação ao sentido da força aplicada [2]. Se o corpo se move para direita com uma força apontando para a esquerda e a velocidade do corpo é maior que zero, então o corpo sofre uma frenagem pois os sentidos dos vetores força e velocidade são contrários. Se o corpo se move para esquerda com uma força para esquerda, e a velocidade é maior que zero, o corpo sofre uma aceleração pois os vetores velocidade e força apontam para o mesmo sentido.

Se houver uma força constante atuando sobre este corpo, então a aceleração gerada por essa força é constante. Ao aplicar uma força resultante que mude de intensidade no decorrer do tempo haverá uma aceleração variando de acordo com a força, pois para toda e qualquer força, a massa do corpo independe do movimento na física mecânica [2].

Ao definir a força constante no simulador de movimentos em duas dimensões no XNA é possível tratá-la através da entrada do jogador para fazer com que essa força seja aplicada a determinado objeto, mudando assim seu vetor velocidade.

4.3.3. Ação e Reação. Uma força gerada pela reação do choque entre esferas muda o sentido da que causou a ação mantendo a direção e a intensidade se considerarmos que o sistema não apresenta forças externas [2]. Todo corpo atingido por outro com determinada força reage com força de mesma intensidade no sentido contrário.

Através da ação e reação, desprezando forças externas, é possível obter colisões elásticas como o choque entre duas esferas de bilhar.

4.4. Força de Atrito

A resistência que atua sobre um corpo que está deslizando sobre uma superfície denomina-se força de atrito cinético [2], definida pela expressão abaixo:

$$\vec{f}_{at} = m * \vec{N}$$

Onde m é o coeficiente de atrito da superfície, portanto, quanto mais lisa é a superfície em que o objeto desliza menor será a força de atrito [2] e \vec{N} é a força normal, perpendicular à força de atrito, fazendo

com que a força de atrito seja proporcional ao peso total do conjunto [2].

Em XNA é possível definir a força de atrito, ou suas componentes, para que os objetos interajam com o ambiente.

5. Resultados

Foram construídos dois simuladores físicos que utilizam os conceitos abordados neste artigo. O primeiro deles apresenta diversas aplicabilidades nas simulações de situações reais em duas dimensões, como resistência à força de atrito, movimento acelerado, movimento retardado e as três leis de Newton.

O componente principal deste simulador é um *sprite* que obtém velocidade através de entradas dos usuários. Cada tecla definida no algoritmo implica em uma força para que o *sprite* possa percorrer o plano cartesiano, parando apenas com a ação da força de atrito do ambiente ou outras entradas de usuário com o sentido da força em oposição ao deslocamento corrente.

A figura 1 apresenta o fluxograma do *Game Loop* para as entradas de usuário no simulador 1.

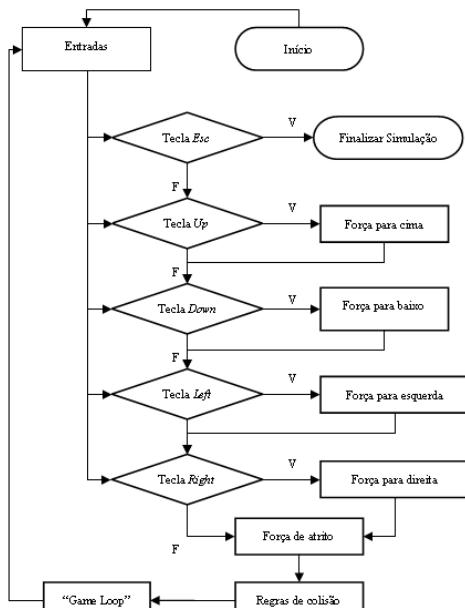


Figura 1. Fluxograma do algoritmo de game loop para o simulador 1.

O segundo simulador é um jogo completo de “pong”, que consiste em um jogo de tênis em duas dimensões. Este simula todas as situações físicas de velocidade e efeito ocasionado pelo choque, definindo a direção resultante de uma esfera. A figura 2 mostra uma seqüência de imagens do simulador 2.

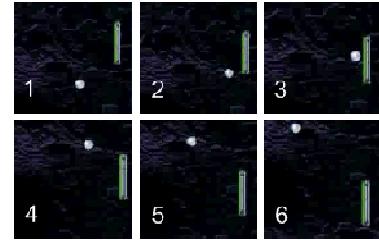


Figura 2. Seqüência de imagens representando o choque elástico no simulador 2.

Na seqüência de imagens pode-se constatar que a trajetória inicial descrita pela esfera, antes do choque com o anteparo, representa uma função linear que cresce no eixo x e no eixo y a cada repetição do *Game Loop*.

Quando a esfera entra em contato com o anteparo verifica-se a relação de forças existentes entre o anteparo e a esfera, (o atrito entre os dois objetos e a força resultante do movimento do anteparo são levados em consideração no momento do choque), causando uma mudança na trajetória da esfera.

6. Conclusão

Este trabalho se dedicou ao estudo da plataforma XNA, a simulação de movimentos físicos em duas dimensões, englobando força de aceleração, força de atrito, inércia, choques elásticos e alguns conceitos matemáticos como, o cálculo do módulo, a normalização, a multiplicação, a soma e a subtração de vetores.

Como resultado foram edificados dois simuladores que aplicam os processos físicos estudados na plataforma XNA, com resultados satisfatórios.

Como trabalho futuro, pretende-se desenvolver simulações em ambientes de três dimensões utilizando a física e matemática como ferramenta de auxílio.

7. Referências

- [1]Dunn, F., Parberry, I., *3D math primer for graphics and game development*, Wordware Publishing, 2002.
- [2]Young, H.D., Freedman, R.A., Sears e Zemansky *Física I Mecânica*, 10 ed São Paulo, Addison Wesley, 2004.
- [3]Tremblay, C., *Mathematics for game Developers*, Thomson, Boston, 2004.
- [4]Lobão, A., Evangelista, B., Farias, J.A.L., *Beginning XNA 2.0 Game Programming*, Apress, New York, 2008.
- [5]Cawood, S., McGee, P., *Microsoft XNA Game Studio Creator's Guide*, McGraw Hill, 2007

Iluminação de Cartoons Baseada em uma Árvore de Regiões

Allan Carlos Rocha
Instituto de Computação-UFAL
allancarlosar@gmail.com

Renata Thomaz Lins
Instituto de Computação-UFAL
renatinhalins@gmail.com

Leandro Carnaúba
Instituto de Computação-UFAL
leandrocarnaubaleite@gmail.com

Adelailson Peixoto (orientador)
Instituto de Matemática-UFAL
adelailson@pos.mat.ufal.br

Vinícius Mello (orientador)
Instituto de Matemática-UFAL
vinicius.moreira.mello@gmail.com



Figura 1. Da esquerda: Cartoon Original, Esqueletização, Mapa de Normais e Cartoon Renderizado.

Resumo

Este trabalho apresenta uma técnica para iluminar cartoons 2D baseada em uma árvore de regiões que contém informações geométricas e topológicas dos objetos. Cada região é composta por um conjunto de pixels internos e um conjunto de curvas. Esta estrutura permite rapidamente a busca de relações de adjacência e inclusão entre regiões, o que facilita operações como colorização e iluminação durante a animação de cartoons. A partir desta estrutura, o trabalho apresenta um método de iluminação que permite o cálculo preciso das normais dos objetos.

1. Introdução

Três importantes etapas na produção de uma seqüência de cartoons são *animação*, *colorização* e *iluminação*. A primeira calcula os quadros intermediários da seqüência a partir do conjunto de quadros iniciais. A colorização transfere as cores de um quadro para os quadros subsequentes. A iluminação calcula a interação do cartoon com as luzes presentes em um ambiente 3D. Este trabalho apresenta um método de iluminação calculado a partir de uma *árvore de regiões* que representa o cartoon. Esta árvore possibilita acesso rápido às relações de adjacência e de inclusão entre regiões, o que permite o cálculo da interpolação das normais no interior de cada região durante a etapa de iluminação.

Alguns trabalhos empregam estruturas hierárquicas que permitem agrupar regiões em diferentes escalas e obter representações da imagem em diversas resoluções [2, 4, 3],

ideais para aplicações como segmentação e compactação de imagens e vídeos. Como o alvo do método aqui apresentado são imagens cartoon, a estrutura empregada para representar as regiões pode ser simplificada, uma vez que o cálculo da iluminação apresentado requer apenas as relações de adjacência e inclusão das regiões. Outros trabalhos empregam estruturas ainda mais simples para segmentar as regiões [5, 11, 12, 1] e realizar operações como transferência de cor durante a animação. Embora estas técnicas funcionem bem na colorização de uma animação de cartoons, as representações de regiões contêm informação suficiente para realizar a iluminação, sendo necessário, para isto, recorrer a outros métodos de iluminação como [7].

Algumas técnicas para iluminar objetos 2D fornecem ferramentas interativas para reconstruir modelos 3D a partir das informações 2D disponíveis [6, 13] e em seguida aplicam técnicas tradicionais de iluminação [9]. Embora estas técnicas produzam bons resultados, elas não são adequadas para cartoons, já que qualquer alteração no número de linhas do desenho pode alterar significativamente a geometria do objeto 3D. Assim, Johnston [7] apresenta uma técnica para aproximar as normais diretamente no próprio desenho do cartoon, sem a necessidade de reconstrução 3D do modelo. Ao contrário de Johnston, o método de iluminação aqui proposto dispõe de um conjunto de informações topológicas e geométricas armazenadas na árvore de regiões que permite o cálculo preciso do mapa de normais. No caso de se ter uma região circular, será demonstrado matematicamente na seção 3.1 que a reconstrução do mapa de normais aqui apresentada corresponde exatamente aos vetores normais de uma esfera.

2. Árvore de Regiões

Essa seção descreve uma estrutura topológica que contém a topologia e geometria dos objetos presentes no cartoon. Assim como em outros métodos, aplica-se inicialmente uma segmentação para extrair um conjunto de curvas e regiões [1]. Em um estágio adicional, foram exploradas outras informações que serão utilizadas para a construção da árvore de regiões. Em [1] tem-se uma descrição completa do conjunto de operações usadas para construir uma estrutura topológica típica. Essas operações incluem: esqueletonização e detecção de regiões.

Esqueletonização A esqueletonização ou *thinning* procura extrair um conjunto de pontos redundantes em uma imagem. Os pontos que não serão removidos constituem o *esqueleto* do objeto. A idéia principal é simplificar a representação do objeto, mantendo sua topologia e a conectividade das linhas. Neste trabalho, foi aplicado o algoritmo de *Zhang-Suen* [14].

Detecção de Regiões Esse estágio procura segmentar a imagem resultante do *thinning*, identificando todas as re-

giões que a compõem. Cada região é definida pelo conjunto de curvas (bordo e curvas internas) e pelo conjunto de pixels internos à região. Estes dois conjuntos são calculados separadamente. Primeiramente são calculados todos os pixels internos a cada região e em seguida aplica-se um rótulo a cada pixel, como uma cor ou um caracter, que identifica a região. Assim, a imagem pode ser vista como um conjunto de regiões.

Numa próxima etapa os pixels que definem o bordo e as curvas internas a cada região R devem ser ordenados para formar o conjunto de curvas poligonais de R . Embora as curvas internas não sejam necessárias para a definição de uma região, elas são importantes, pois contêm características relevantes. Em desenhos artísticos, por exemplo, as curvas internas podem fornecer uma variedade de detalhes que são diretamente responsáveis por dar um efeito mais realístico ao desenho. Para obter uma representação poligonal das curvas de cada região, pode-se aplicar o algoritmo de *chain-code* com vizinhança 8-conectado [8]. Após esta etapa cada região possuirá: uma área interna (conjunto de pixels rotulados), uma curva de bordo e um conjunto de curvas internas.

2.1. Geração da Árvore de Regiões

Estruturas topológicas usuais utilizadas em alguns métodos de colorização ajudam a rastrear regiões correspondentes de quadros consecutivos da animação [5, 11, 12, 1]. Tais estruturas exploram basicamente as relações de adjacência entre as regiões para manter essa coerência temporal, porém elas não exploram suficientemente as informações locais que estão em um mesmo quadro, como as relações de inclusão entre regiões do desenho. Assim sendo, a estrutura aqui descrita engloba a relação de subconjunto entre as regiões e explora esse fato, por exemplo, para aplicar novas operações no cartoon, como a iluminação apresentada na seção 3.

A estrutura aqui proposta é definida como uma árvore, onde cada nó contém uma região R_i e um conjunto de nós descendentes que representam as regiões internas (contidas) à R_i . Além de ser uma estrutura simples e fácil de ser implementada, ela contém mais informações do que as estruturas topológicas usuais (Figura 2).

A árvore de regiões proposta provê o controle total no acesso à cada região: curva de bordo e curvas internas, pixels interiores, regiões adjacentes e regiões contidas. Isso permite que sejam definidos atributos separadamente a cada região do cartoon como: propriedade dos materiais, cor, transparência, textura, etc. Da mesma forma, podem ser definidos atributos para cada curva como: cor, estilo de linha, espessura, etc. Os atributos das regiões podem ser armazenados em um frame buffer, numa estratégia inspirada no OpenGL [10], o que permite que os dados sejam proce-

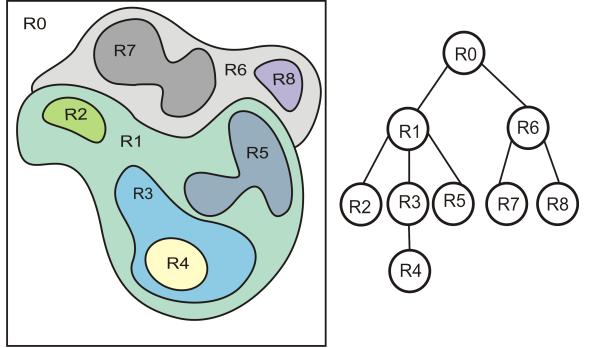


Figura 2. Hierarquia de Regiões

sados na GPU. No contexto deste trabalho, as normais são os atributos mais importantes relacionados às regiões ou curvas. A próxima seção trata desses atributos.

3. Iluminação

Esta seção apresenta uma técnica para calcular o mapa de normais baseada na árvore de regiões proposta. O cálculo das normais é realizado primeiro nas curvas de cada região e em seguida as normais são interpoladas nos pixels internos de cada região. Como as curvas de cada região estão ordenadas, as normais de cada curva foram calculadas como em [1]: dados dois pontos consecutivos $p_i = (x_i, y_i)$ e $p_{i+1} = (x_{i+1}, y_{i+1})$ da curva, calcula-se a normal como $n_i = (y_{i+1} - y_i, x_i - x_{i+1}, 0)$. A coordenada $z = 0$ indica que o plano da imagem é o plano de projeção em relação ao observador e que a curva faz parte da silhueta do objeto. Como há apenas oito possíveis direções para o vetor n_i (já que foi usada a vizinhança 8-conectado) é possível suavizar o mapa de normais, substituindo n_i por uma média ponderada entre n_i e as normais de seus vizinhos: $n_i = (n_{i-2} + 4n_{i-1} + 6n_i + 4n_{i+1} + n_{i+2})/16$.

3.1. Interpolação das Normais

Aqui é apresentada a idéia básica empregada no método de iluminação proposto. Seja C uma curva fechada parametrizada pelo comprimento de arco, ou seja, $C(s) = (x(s), y(s))$, e seja R a região delimitada por C . A normal $n(s)$ em cada ponto $C(s)$ é dada por $n(s) = (n_x(s), n_y(s)) = (y'(s), -x'(s))$. O objetivo é calcular uma normal 3D $n(p) = (n_x(p), n_y(p), n_z(p))$ para cada ponto p em R . Para calcular as componentes x e y de $n(p)$, integra-se cada contribuição da normal ao longo da curva, utilizando um peso proporcional ao quadrado do inverso da distância de p a $C(s)$. Assim, tem-se

$$n_x(p) = \frac{\int_C \frac{n_x(s)ds}{|p - C(s)|^2}}{w(p)},$$

$$n_y(p) = \frac{\int_C \frac{n_y(s)ds}{|p - C(s)|^2}}{w(p)},$$

onde

$$w(p) = \int_C \frac{ds}{|p - C(s)|^2}.$$

De modo a manter o vetor normal unitário, a componente z é definida como

$$n_z(p) = \sqrt{1 - n_x(p)^2 - n_y(p)^2}.$$

Uma análise desta fórmula quando C é um círculo, isto é, $C(s) = (\cos s, \sin s)$ mostra que as normais reconstruídas equivalem às normais de uma esfera: graças à simetria radial, será considerado apenas o caso $p = (x, 0)$, com $-1 < x < 1$ (caso contrário pode-se simplesmente girar o sistema de coordenadas). Tem-se que $n_y(p)$ deve ser zero, pois a contribuição de y de cada $C(s)$ é cancelada pela contribuição de $C(-s)$. Resolvendo a integração com ajuda de um sistema de álgebra computacional, tem-se

$$w(p) = \int_0^{2\pi} \frac{ds}{(x - \cos s)^2 + \sin^2 s} = \frac{2\pi}{1 - x^2},$$

$$N_x(p) = \int_0^{2\pi} \frac{\cos s ds}{(x - \cos s)^2 + \sin^2 s} = \frac{2\pi x}{1 - x^2}.$$

Como $N_y(p) = 0$ por simetria, $N_z(p) = \sqrt{1 - x^2}$. Assim, percebe-se que o campo de normal é exatamente o mesmo de uma esfera de raio 1 centrada na origem.

Diferentemente de [7], que faz uma propagação aproximada para o cálculo das normais, a formulação aqui explícita é precisa. Além disso, devido à flexibilidade da estrutura proposta, é possível escolher quais curvas podem contribuir na interpolação de cada região e obter os efeitos propostos em [7] (Figura 3).



Figura 3. Diferentes Cálculos das Normais

A orientação das normais das curvas também pode ser controlada para obter outros resultados. Como cada região e curva da estrutura topológica aqui apresentada pode ser acessada independentemente, pode-se definir e aplicar diversos operadores aos atributos. Foram implementados dois operadores de normais: *operador de escala* e o *operador de profundidade*. Dada uma curva C , o operador de escala consiste de uma escala na coordenada z das normais do atributo selecionado (Figura 4).



Figura 4. Operador de Escala

O operador de profundidade trabalha com a idéia de levantar/afundar uma dada curva C em uma região. O usuário controla os efeitos do operador selecionando uma curva desejada (Figura 5).



Figura 5. Operador Profundidade

As Figuras 1 e 6 mostram alguns resultados.



Figura 6. Cartoon Renderizado

4. Conclusões e Trabalhos Futuros

Este trabalho apresentou um método de iluminação de cartoons baseado em uma estruturação do cartoon como uma árvore de regiões. Diferentemente de outras estruturas, a representação aqui proposta explora a coerência espacial local de cada frame, verificando relações de subconjunto de regiões. Além disso mantém todas as informações de vizinhança presentes nas estruturas topológicas usuais. O método de iluminação proposto realiza o cálculo preciso do

mapa de normais e apresenta uma grande flexibilidade na obtenção de diferentes efeitos na visualização do cartoon.

Etapas futuras incluem: verificar a estrutura em cartoons animados; definir novos operadores de atributos; verificar quais curvas de cada região são invisíveis a um ponto interno, de modo a evitar sua contribuição no cálculo do vetor normal ao ponto; utilizar outras estruturas de dados mais eficientes para otimizar o cálculo do mapa de normais; permitir que as operações de atributos sejam enviadas e processadas na GPU a fim de acelerar o processamento.

Referências

- [1] H. Bezerra, L. Velho, and B. Feijó. A computer-assisted colorization algorithm based on topological difference. *XIX Brazilian Symposium on Computer Graphics and Image Processing*, pages 121–127, 2006.
- [2] M. De Carvalho, M. Couprise, and R. De Alencar Lotufo. Image segmentation by analysis of scale-space. *Computer Graphics and Image Processing, 2002. Proceedings. XV Brazilian Symposium on*, pages 403–, 2002.
- [3] M. de Carvalho and R. de Alencar Lotufo. Hierarchical regions matching in image sequences through association graph. *Computer Graphics and Image Processing, 2001 Proceedings of XIV Brazilian Symposium on*, pages 396–, Oct 2001.
- [4] L. Garrido, P. Salembier, and J. R. Casas. Representing and retrieving regions using binary partition trees. In *In IEEE International Conference on Image Processing, ICIP 99*, 1999.
- [5] S. S. Hock and F. Tian. Computer-assisted coloring by matching line drawings. *The Visual Computer*, pages 289–304, 2000.
- [6] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3d freeform design. *Computer Graphics*, pages 409–416, 1999.
- [7] S. F. Johnston. Illumination for cel animation. *NPAR - 2RD International Symposium on Nonphotorealistic Animation and Rendering*, pages 45–FF, 2002.
- [8] J. Parker. *Practical Computer Vision using C*. John Wiley and Sons, Inc, New York, 1994.
- [9] B. T. Phong. Illumination for computer generated pictures. *Graphics and Image Processing*, pages 311–317, 1975.
- [10] D. Shreiner, M. Woo, J. Neider, and T. Davis. *OpenGL(R) Programming Guide : The Official Guide to Learning OpenGL(R), Version 2 (5th Edition)*. 2005.
- [11] D. Sykora, J. Burianek, and J. Zara. Segmentation of black and white cartoons. *Spring Conference on Computer Graphics*, pages 245–254, 2003.
- [12] D. Sykora, J. Burianek, and J. Zara. Unsupervised colorization of black-and-white cartoons. *NPAR - 2RD International Symposium on Nonphotorealistic Animation and Rendering*, pages 121–127, 2004.
- [13] R. C. Zeleznik, K. P. Herndon, and J. F. Hud Hess. Sketch: an interface for sketching 3d scenes. *Computer Graphics*, pages 163–170, 1996.
- [14] Zhang and Suen. *A fast parallel algorithm for thinning digital patterns*. Comm, ACM, 1984.

Deformação de Objetos 2D Utilizando Curvas de Bézier e Mínimos Quadrados Móveis

Douglas Cedrim
Instituto de Computação - UFAL
{dco@tci.ufal.br}

Adelailson Peixoto (Orientador)
Instituto de Matemática - UFAL
{adelailson@pos.mat.ufal.br}

Resumo

Este trabalho apresenta um método para deformar espacialmente objetos 2D através da manipulação de curvas de Bézier. O objetivo é aumentar o controle do usuário sobre a deformação. O trabalho emprega a abordagem proposta por [8] de mínimos quadrados móveis para computar as deformações e aplicá-las na imagem a uma complexidade polinomial.

1. Introdução

Alguns dos principais desafios enfrentados na deformação de imagens estão ligados à especificação da deformação, que requer ferramentas que dêem ao usuário um controle real sobre a deformação [6, 9]. A partir da especificação os algoritmos de deformação são aplicados aos objetos. No caso de imagens [11, 2, 8], a especificação inclui pontos, segmentos de retas, curvas, regiões e outros. Matematicamente, uma deformação pode ser definida como uma transformação que leva pontos de um objeto gráfico a outro. Este trabalho tem como principal objetivo deformar imagens especificadas a partir de curvas. Algumas aplicações de deformação estão ligadas à animação de *cartoons*, onde os objetos-chave, definidos por deformações, são interpolados para criar transições suaves entre eles [4]. Na área médica [3] as deformações são utilizadas na análise e reconstrução de modelos a partir de dados de tomografia, ressonância magnética. Outras aplicações estão ligadas ao *Morphing* de objetos [10], entretenimentos e outros.

Diversos trabalhos calculam deformações de imagens através de estruturas auxiliares, sobre as quais a deformação é especificada [11, 2, 6]. Em [6] é utilizada uma estrutura quadtree sobre a qual a deformação é calculada e então repassada aos objetos através de funções de base radial. O cálculo emprega a minimização da energia entre cada duas células vizinhas da quadtree. Apesar de apresentar bons resultados, em geral estes métodos exigem muito

esforço na implementação de diversos algoritmos. Em [8] as deformações são especificadas por pontos e linhas, em seguida seu cálculo emprega mínimos quadrados móveis para computar as novas posições dos pontos da imagem. A abordagem aqui apresentada, porém, estende a especificação de [8] para *splines de Bézier*, dando ao usuário um maior controle sobre a deformação.

2. Fundamentação

Esta seção traz uma breve abordagem sobre curvas de Bézier [7] e em seguida serão vistos os *mínimos quadrados móveis* e sua extensão com linhas [8].

2.1. Curvas de Bézier

Dados k pontos P_0, P_1, \dots, P_k , uma curva paramétrica c de Bézier de grau $k - 1$ é descrita como

$$c(t) = \sum_{i=0}^k P_i b_{k,i}(t), \quad (1)$$

onde $b_{k,i}$ é o i -ésimo coeficiente polinomial de Bernstein, definido como segue

$$b_{k,i}(t) = \binom{k}{i} t^i (1-t)^{k-i} \quad (2)$$

2.2. Mínimos Quadrados Móveis

Seja p um ponto de uma imagem I e sejam x_i, y_i pontos não deformados e deformados, respectivamente, especificados em I . Para cada ponto p , deve-se encontrar a função de deformação apropriada f , que irá mapear os pontos de controle x_i nos pontos deformados y_i e utilizá-la para deformar p . Ou seja, f deve minimizar a função ponderada

$$\sum_i w_i \|f(x_i) - y_i\|^2 \quad (3)$$

Como f é computada para cada $p \in I$ e a influência dos pontos de controle x_i varia de acordo com a distância, o peso w_i deve ser dependente do ponto avaliado ¹.

Assim, w_i aumenta à medida que p se aproxima dos pontos x_i . Em [8], o peso w_i é dado por

$$w_i = \|x_i - p\|^{-\alpha},$$

sendo suave para qualquer $\alpha \geq 2$.

Como $f_m(p)$ é uma transformação afim, ou seja, $f_m(p) = pM + T$, podemos reescrever (3) em sua forma matricial, o que leva à equação

$$\min \sum_i w_i \|(x_i M + T) - y_i\|^2 \quad (4)$$

Resolvendo a equação acima para T, tem-se

$$T = y_c - x_c M,$$

logo

$$\min \sum_i w_i \|\hat{x}_i M - \hat{y}_i\|^2, \quad (5)$$

onde $\hat{x}_i = x_i - x_c$ e $\hat{y}_i = y_i - y_c$ com

$$x_c = \frac{\sum_i w_i x_i}{\sum_i w_i},$$

$$y_c = \frac{\sum_i w_i y_i}{\sum_i w_i}$$

2.3. Mínimos Quadrados Móveis com Segmentos de Linha

Em [8] Schaefer estende o controle da deformação para segmentos de retas $x_i(t)$, $y_i(t)$, em vez de pontos. Assim, o problema de minimização passa a ser

$$\sum_i \int_0^1 w_i(t) \|\hat{x}_i(t) M - \hat{y}_i(t)\|^2,$$

onde a integral representa um grande conjunto de pontos de amostra,

$$w_i(t) = \|x'_i(t)\| (\|x_i(t) - p\|)^{-2\alpha}$$

representa uma função de peso modificada que é independente da parametrização da curva e

$$\hat{x}_i(t) = x_i(t) - x_c, \quad \hat{y}_i(t) = y_i(t) - y_c$$

são as curvas de controle transladas pelos centróides definidos a seguir

¹ Ver [5] para maiores detalhes.

$$x_c = \frac{\sum_i \int_0^1 w_i(t) x_i(t) dt}{\sum_i \int_0^1 w_i(t) dt}$$

$$y_c = \frac{\sum_i \int_0^1 w_i(t) y_i(t) dt}{\sum_i \int_0^1 w_i(t) dt}$$

3. Método

Aqui, a idéia de [8] é estendida para curvas de controle genéricas. As curvas de controle a serem usadas na deformação são definidas como:

$$\hat{x}_i(t) = \begin{pmatrix} 1-t & t \end{pmatrix} \begin{pmatrix} \hat{a}_i \\ \hat{b}_i \end{pmatrix}$$

$$\hat{y}_i(t) = \begin{pmatrix} 1-t & t \end{pmatrix} \begin{pmatrix} \hat{c}_i \\ \hat{d}_i \end{pmatrix}$$

onde $(\hat{a}_i, \hat{b}_i) = (a_i - x_c, b_i - y_c)$ são os pontos inicial e final da curva não deformada devidamente transladados e $(\hat{c}_i, \hat{d}_i) = (c_i - x_c, d_i - y_c)$ os pontos inicial e final da curva deformada.

Inicialmente, resolve-se para uma família mais geral de matrizes, contendo todo tipo de transformação afim, para então desenvolver transformações mais específicas e resultados visuais melhores. Como pode-se ver em [5], a transformação varia de acordo com o ponto avaliado.

3.1. Deformações Afins

A solução do problema inicial de minimização, mais especificamente da equação (5), será determinada pelos pontos críticos, obtendo assim a matriz que minimiza o quadrado dos desvios de cada deformação. Através disso, obtém-se uma matriz geral bem definida para cada ponto p pertencente à imagem, que irá representar a transformação afim que é solução do problema para aquele ponto. Dessa forma, expandindo (5) e resolvendo para M , tem-se

$$M_p = \left(\sum_i \hat{x}_i^T w_i \hat{x}_i \right)^{-1} \left(\sum_j \hat{x}_j^T \hat{y}_j w_j \right),$$

onde \hat{x}_i são os pontos de controle não deformados e \hat{y}_i os deformados.

Uma vez que foco não está no uso de pontos como controladores da deformação, mas sim curvas de Bézier, elas cumprem esse papel de melhorar o controle sobre a deformação. Assim, utilizando a definição de curvas através da multiplicação de dois vetores (3), as equações são

$$\sum_i \int_0^1 \left\| \begin{pmatrix} 1-t & t \end{pmatrix} \left(\begin{pmatrix} \hat{a}_i \\ \hat{b}_i \end{pmatrix} M - \begin{pmatrix} \hat{c}_i \\ \hat{d}_i \end{pmatrix} \right) \right\|^2, \quad (6)$$

onde, resolvendo para M , teremos

$$M_a = (M_1)^{-1} \sum_j \begin{pmatrix} \hat{a}_j \\ \hat{b}_j \end{pmatrix}^T W_j \begin{pmatrix} \hat{c}_j \\ \hat{d}_j \end{pmatrix}, \quad (7)$$

com

$$M_1 = \left(\sum_i \begin{pmatrix} \hat{a}_i \\ \hat{b}_i \end{pmatrix}^T W_i \begin{pmatrix} \hat{a}_i \\ \hat{b}_i \end{pmatrix} \right) \quad (8)$$

onde

$$W_j = W_i = \begin{pmatrix} w_{i,0} & w_{i,1} \\ w_{i,1} & w_{i,2} \end{pmatrix}$$

sendo cada w_i a avaliação das integrais

$$w_{i,0} = \int_0^1 w_i(t)(1-t)^2 dt$$

$$w_{i,1} = \int_0^1 w_i(t)(1-t)dt$$

$$w_{i,2} = \int_0^1 w_i(t)t^2 dt$$

Os centróides para os segmentos de linhas x_c, y_c podem ser calculados utilizando essas integrais. Assim

$$x_c = \frac{\sum_i a_i(w_{i,0} + w_{i,1}) + b_i(w_{i,1} + w_{i,2})}{\sum_i w_{i,0} + 2w_{i,1} + w_{i,2}}$$

$$y_c = \frac{\sum_i c_i(w_{i,0} + w_{i,1}) + d_i(w_{i,1} + w_{i,2})}{\sum_i w_{i,0} + 2w_{i,1} + w_{i,2}}$$

Uma vez que trata-se de uma família geral de transformações afins, cisalhamento e escala não uniforme fazem parte dessas transformações, não preservam a rigidez da deformação, consequentemente seu realismo.

3.2. Deformações por Similaridade

A partir das transformações afins, obtém-se outra família mais restrita de matrizes de forma a eliminar o cisalhamento das imagens, tornando a deformação quase rígida, a menos de uma escala não uniforme ([8]). Assim é determinada uma transformação afim com a propriedade de que sua matriz M , seja tal que $M^T M = \lambda^2 I$, onde λ é um real. Para determinar a matriz, é preciso apenas calcular sua primeira coluna $v1 = (a, b)$, de forma que, $\langle v1, v2 \rangle = 0 \Rightarrow v2 = (-b, a) = v1^\perp$. Seja a matriz

$$M' = [v1 \ v1^\perp]$$

Assim teremos

$$\min \sum_i \int_0^1 \left\| A(t) \left(\begin{pmatrix} \hat{a}_i \\ -\hat{a}_i^\perp \\ \hat{b}_i \\ -\hat{b}_i^\perp \end{pmatrix} M' - \begin{pmatrix} \hat{c}_i^T \\ \hat{d}_i^T \end{pmatrix} \right) \right\|^2, \quad (9)$$

com

$$A(t) = \begin{pmatrix} 1-t & 0 & t & 0 \\ 0 & 1-t & 0 & t \end{pmatrix}$$

onde, resolvendo para M' , passaremos a ter,

$$M_s = \frac{\sum_i \begin{pmatrix} \hat{a}_i \\ -\hat{a}_i^\perp \\ \hat{b}_i \\ -\hat{b}_i^\perp \end{pmatrix}^T W_i \begin{pmatrix} \hat{c}_i^T & -\hat{c}_i^{\perp T} \\ \hat{d}_i^T & -\hat{d}_i^{\perp T} \end{pmatrix}}{\mu_s} \quad (10)$$

com

$$\mu_s = \sum_i \|\hat{a}_i\|^2 w_{i,0} + 2 \langle \hat{a}_i, \hat{b}_i \rangle w_{i,1} + \|\hat{b}_i\|^2 w_{i,2}$$

e a matriz de ordem 4 dos pesos ([8]) definida como

$$W_i = \begin{pmatrix} w_{i,0} & 0 & w_{i,1} & 0 \\ 0 & w_{i,0} & 0 & w_{i,1} \\ w_{i,1} & 0 & w_{i,2} & 0 \\ 0 & w_{i,1} & 0 & w_{i,2} \end{pmatrix}$$

3.3. Deformações Rígidas

Restringir as transformações afins para que não possuam nem cisalhamento, nem escala não uniforme equivale a minimizar (9) sujeito a $M^T M = I$. Esse problema assemelha-se ao anterior, e pode ser resolvido de forma análoga a menos da mudança do fator de escala μ_s para um

$$\mu_r = \left\| \sum_i \begin{pmatrix} \hat{a}_i^T & -\hat{a}_i^{\perp T} & \hat{b}_i^T & -\hat{b}_i^{\perp T} \end{pmatrix} W_i \begin{pmatrix} \hat{c}_i^T \\ \hat{d}_i^T \end{pmatrix} \right\|$$

Assim, não é necessário efetuar decomposições de auto-vetores para determinar a melhor transformação [1].

4. Resultados

A figura 1 mostra a especificação e a deformação rígida de uma garrafa. A figura 2 deforma localmente o olho do falcão. A figura 3 mostra a relação entre o número de amostras da curva e o tempo de processamento.



Figura 1. Deformação de uma Garrafa.



Figura 2. Deformação de um Falcão.

5. Conclusões e Trabalhos Futuros

Este trabalho apresentou uma ferramenta de deformação de imagens utilizando curvas de Bézier para especificar a deformação. O método, baseado em [8], emprega mínimos quadrados móveis para calcular as novas posições dos pixels da imagem a partir da especificação. O uso de curvas permite ao usuário ter um maior controle ao especificar a deformação da imagem, tornando-a mais intuitiva. Como trabalhos futuros temos: a manipulação direta das curvas splines durante a especificação da deformação, em vez de manipular seus pontos de controles; o desenvolvimento de outras ferramentas para deformar a imagem; um estudo de técnicas para impedir dobras da grade de modo a evitar sobreposição de pixels durante a deformação da imagem.

Referências

- [1] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. In *J. Opt. Soc. Am. A* 4, page 629, 1987.
- [2] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 1134–1141, New York, NY, USA, 2005. ACM.
- [3] T. Ju, J. Warren, G. Eichele, C. Thaller, W. Chiu, and J. Carlson. A geometric database for gene expression data. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 166–176, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [4] A. Kort. Computer aided inbetweening. In *NPAR '02: Proceedings of the 2nd international symposium on Non-*

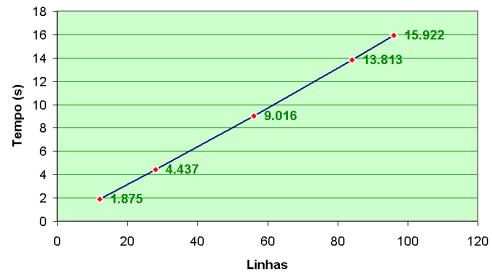


Figura 3. Amostras em Função do Tempo.

photorealistic animation and rendering, pages 125–132, New York, NY, USA, 2002. ACM.

- [5] A. Nealen. An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation. 2004.
- [6] A. Nealen. Adaptive space deformations based on rigid cells. *EUROGRAPHICS Comput. Graph.*, 26(3), 2007.
- [7] L. Piegl and W. Tiller. *The NURBS Book*. Springer, 2 edition, 1996.
- [8] S. Schaefer, T. McPhail, and J. Warren. Image deformation using moving least squares. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 533–540, New York, NY, USA, 2006. ACM.
- [9] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.*, 20(4):151–160, 1986.
- [10] D. B. Smythe. A Two-Pass Mesh warping algorithm for object transformation and image interpolation. Technical report, 1990.
- [11] Y. Weng, W. Xu, Y. Wu, K. Zhou, and B. Guo. 2d shape deformation using nonlinear least squares optimization. *Vis. Comput.*, 22(9):653–660, 2006.

Estimando Cobertura Vegetal por Análise de Imagens

Jaime Daniel Corrêa Mendes
Universidade Federal de Lavras
Departamento de Ciência da Computação
CP3037, Lavras, MG, Brasil
lordshark@comp.ufla.br

André Vital Saúde (Orientador)
Universidade Federal de Lavras
Departamento de Ciência da Computação
CP3037, Lavras, MG, Brasil
saude@dcc.ufla.br

Resumo

Os biocombustíveis têm adquirido grande notoriedade e investimentos por partes governamentais e instituições de pesquisa por apresentarem-se como ótima alternativa aos combustíveis não-renováveis, principalmente devido aos seus benefícios econômicos, sociais, ambientais, tecnológicos e estratégicos. Devido ao fato de a principal matéria-prima ser oriunda de plantas oleaginosas e permitir cultivos em consórcio com outras plantas, essas estão sendo largamente exploradas em sistema de agricultura familiar cujo acesso a novas tecnologias não é muito explorado. Com isso, para aliar necessidade, oportunidade e tecnologia, o presente trabalho propõe uma alternativa para analisar a radiação solar, um dos principais fatores na otimização no uso da terra, através de técnicas de morfologia matemática e interatividade com o usuário. Um dos papéis deste software é auxiliar na tomada de decisão nas atividades de campo.

1. Introdução

O Biodiesel é um combustível renovável alternativo derivado de óleos vegetais, como girassol, mamona, soja, canola e demais oleaginosas, ou de gorduras de animais. As pesquisas desenvolvidas para a produção de Biodiesel requerem uma cooperação de diversas áreas do conhecimento visando a melhorias, principalmente no setor tecnológico, proporcionando maior qualidade e quantidade do produto final [4].

A principal fonte de matéria-prima para o Biodiesel é originada das plantas oleaginosas, produzidas em grandes, médias e pequenas propriedades. Estas, em geral no sistema de agricultura familiar, remetendo um maior esforço para os produtores em atender à resolução da Agência Nacional do Petróleo (ANP) que requer a inclusão de 3% de Biodiesel ao diesel mineral, a partir de julho de 2008, representando uma nova e importante opção de renda para a agricultura e neste

contexto o estado de Minas Gerais se destaca por ser o segundo maior consumidor de óleo diesel do Brasil [6].

Devido a essa demanda e, de acordo com a ordem do dia, exibida pelos principais jornais brasileiros, governos internacionais criticam a política do governo brasileiro cujo investimento, aplicado na produção de Biodiesel em âmbito nacional, deveria ser destinado à produção de alimentos [3].

Uma prática utilizada principalmente nas regiões tropicais é o cultivo consorciado, que pode envolver cereais, leguminosas e oleaginosas potenciais para a produção de Biodiesel. Pelo fato de as leguminosas serem menos competitivas, o consórcio de oleaginosas e leguminosas podem beneficiar tanto a dieta quanto equilibrar a receita econômica do produtor [1].

Para organizar esta competição pelo solo, uma das variáveis a ser trabalhada para melhorar o seu uso é a otimização do aproveitamento de radiação solar, que é a principal fonte de energia para as plantas, além de influenciar diretamente o rendimento da cultura associado ao arranjo das plantas. Assim a área de folhas da cultura é um indicativo de sua capacidade de aproveitamento de radiação solar e, consequentemente, de sua capacidade de produção de assimilados. [8].

A medida de radiação solar geralmente é feita em estações meteorológicas ou com a utilização de satélites, o que deixa claro o difícil acesso a equipamentos para aquisição de dados referentes à radiação, a fim de incrementar a organização de sua cultura para uma melhor produção, quando comparado à renda obtida por pequenos e médios produtores [2, 10].

1.1. Objetivo

O desenvolvimento tecnológico tem disponibilizado acesso a equipamentos de aquisição e processamento de imagens de alto desempenho e facilidade de uso, que leva o presente trabalho a propor uma alternativa à avaliação da radiação solar em uma área de cultivo por processamento de imagens digitais, captadas com câmeras digitais,

o que apresenta uma alternativa aos altos custos das ferramentas de análise de radiação solar.

Para o processamento dessas imagens, este projeto objetiva especificamente: desenvolver um software de processamento de imagens que permite o gerenciamento das informações obtidas no experimento.

2. Material e Métodos

2.1. Material

Os ensaios foram conduzidos no primeiro semestre letivo do ano de 2008 no Setor de Grandes Culturas do Departamento de Agricultura da Universidade Federal de Lavras, através de experimentos conduzidos pelo grupo G-Óleo, compostos de mamona, canola, pinhão manso, nabo forrageiro, crambe e girassol.

A linguagem de programação utilizada foi JAVA da Sun Microsystems, baseado na *Java Development Kit (JDK) 1.6 Update 01 e Update 02*, sobre a *Integrated Development Environment (IDE) Eclipse Classic* na versão 3.3.1. Para o processamento de imagem partiu-se do demo *Supervised Watershed* de Sébastien Lefèvre, disponível em <http://dpt-info.u-strasbg.fr/~lefevre/>, acessado em 22 de março de 2008, e a biblioteca *ImageJ®* na versão 1.38. O IDE NetBeans nas versões 5.5.1 e 6.0 foi utilizado na criação da interface com o usuário e o Sistema Gerenciador de Banco de Dados foi PostgreSQL na versão 8.4.2.

A aquisição das imagens foi feita com câmeras digitais convencionais: Sony® CyberShot® modelo P73 com 4.1 MegaPixels e lente Sony Lens 6 - 18mm; modelo W5 com 5.1 MegaPixels e lente Carl Zeiss Vario-Tessar 7.9 - 23.7mm. Canon® Rebel® modelo XTi EOS com 10.1 MegaPixels e lente Canon EF Lens com controle remoto. Todas as câmeras utilizaram como suporte um tripé Velbon® modelo CX 200.

2.2. Métodos

O método aplicado para capturar as imagens em campo é feito acoplando as câmeras digitais no tripé, mantendo a lente sempre perpendicular às folhas da cultura. O posicionamento do tripé é feito de acordo com a altura que capture, da melhor forma possível, a área de interesse de análise do usuário em relação à altura atingida pela cultura, sendo irrelevantes as pequenas inclinações que possam ocorrer durante a captura da imagem devido ao movimento aparente do sol que faz com que o ângulo de incidência de radiação varie ao longo do dia. As imagens utilizadas neste trabalho foram captadas posicionando o tripé no centro da rua, com o intuito de registrar a área onde a radiação atinge diretamente o solo e seja possível ver o túnel de crescimento do dossel vegetativo. Para as máquinas sem

controle remoto, as imagens são obtidas ativando o temporizador da máquina com tempo suficiente para erguer o tripé. A Figura 1 mostra como foi feita a aquisição destas imagens em campo.



Figure 1. Método utilizado para capturar as imagens em campo.

Como a medição da área em uma imagem é feito pela contagem de pixels, devido à diferença entre as lentes das máquinas e suas resoluções, e para que a altura do posicionamento do tripé não interfira nas medidas, um teste de prova é utilizado na aquisição de imagens, colocando uma folha A4 (padrão ISO com dimensões de 21 x 29,7 cm) no solo, como objeto de referência, com suas bordas paralelas às linhas da cultura. A calibração das câmeras foi ajustada de acordo com o dia, através das macros das câmeras ou configuração manual, evitando que quantidade de luz interfira no contraste entre plantas e solo.

Ao descarregar as imagens da câmera, elas serão manipuladas por um software que faz o processamento das imagens e registra o experimento no banco de dados como uma descrição dos passos utilizados.

Para medir a área de incidência de radiação solar na cultura, foi convencionado que essa área corresponde à quantidade de pixels correspondente ao solo visível.

Com o intuito de segmentar estas regiões de pixels, o usuário utilizou uma interface interativa do software desenvolvido, em conjunto com operadores básicos de processamento digital de imagens [7] e de morfologia matemática [5], inserindo marcadores de acordo com a área de interesse de sua pesquisa, como mostra a Figura 2 e o resultado da operação na Figura 3.

Com o intuito de obter uma boa relação do tamanho dos pixels da imagem com a área real correspondente, a inserção do objeto de referência impede que as variações na altura da câmera interfiram nas dimensões da imagem. Portanto o trabalho consistiu em segmentar o solo e o objeto de referência na imagem e, em seguida, apresentar dados legíveis ao usuário, medir o número de pixels do solo e o número de pixels do objeto de referência, obtendo a área



Figure 2. Inserção dos marcadores para segmentar a imagem e o resultado da operação.

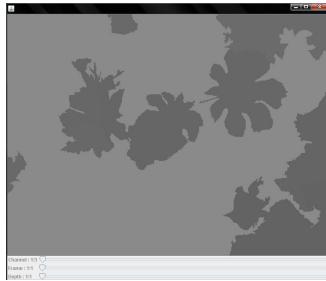


Figure 3. Resultado da Segmentação por Watershed por marcadores da Figura 2.

do solo em pixels. Para obter a área em unidade de medidas reais, foi aplicada a relação:

$$N_{ps} + N_{po} = A_s \quad (1)$$

$$N_{po} = A_{or} \quad (2)$$

onde N_{ps} é o número de pixels de solo visível, N_{po} é o número de pixels do objeto de referência, A_s é a área de solo visível e A_{or} é a área do objeto de referência.

3. Resultados e Discussão

Antes de discutir a técnica utilizada, ressalta-se, principalmente a relação de custos analisadas por este trabalho. Devido ao alto custo de uma imagem de satélite, e mesmo esta possuindo a capacidade de segmentar satisfatoriamente uma área vegetal devido à presença de sensores no equipamento, cujo comprimento de onda as folhas respondem muito bem, outras alternativas de baixo custo podem ser implementadas. O acoplamento da máquina digital em um tripé, deixando a lente perpendicular à área de cultivo apresentou resultados satisfatórios que permitem uma boa relação entre a imagem e a área real, de forma a poder organizar melhor o experimento, implicando a redução dos

custos para permitir o uso da tecnologia em pequenas propriedades agrárias.

3.1. Aquisição e Processamento de Imagens

Visando facilitar a segmentação e evitando trabalhar com imagens coloridas, vários pré-processamentos para realçar os objetos da imagem e tentativas de segmentação, de forma automática, utilizando o ImageJ® foram aplicados nas imagens capturadas. Porém não apresentaram resultados muito satisfatórios, por apresentarem diversos elementos que apenas são separados de forma mais criteriosa, como as pequenas áreas de solo entre as folhas.

Dentre essas tentativas, a separação das componentes no espaço de cor LAB apresentou um excelente resultado de contraste entre folhas e solo no canal a , apresentando uma alternativa direta ao método de segmentação de solo e plantas proposto por [9]. Apesar de não se tratar de um filtro e possuir uma complexidade proporcional ao tamanho da imagem, o custo da transformação para o espaço LAB não consome tanto processamento quanto uma filtragem, mas é considerável devido à precisão dos cálculos aplicados nesta conversão, pelo fato de não existir uma conversão direta do espaço RGB e para o LAB.

A separação de componentes no espaço LAB também se mostrou favorável em um pré-processamento da folha. A folha na cor branca é facilmente identificada na componente b , o que permitiu segmentá-la facilmente, aplicando uma operação de abertura morfológica por um quadrado seguido de $threshold$. Esta operação foi possível pelo fato de as bordas da imagem estarem paralelas às linhas da cultura.

A Figura 4 mostra o destaque da folha em relação ao restante da imagem na componente b da imagem.



Figure 4. Imagem original e realce do objeto de referência (folha), visualizando apenas a componente b .

O uso do espaço de cor LAB é que o alto contraste - entre folhas e solo - permitiu a introdução de métodos automáticos para segmentar a imagem, mas os “espaços vazios”, onde os vegetais se concentram, eram ignorados, o que poderia falhar na corretitude da análise. Para este fim,

foi aplicado o Watershed por marcadores de forma interativa, o que permitiu que as segmentações de pequenas áreas fossem mais precisas e incluídas na medição de área.

A Figura 5 mostra o número de marcadores utilizados em uma das imagens do experimento e o resultado da segmentação



Figure 5. Marcadores em uma das imagens do experimento e o resultado da segmentação.

No que se refere à análise de imagens, os erros de perspectiva e inclinação da câmera não são relevantes pois na quantificação das informações é feita uma média das imagens capturadas.

O software permitiu a organização de um banco de dados de informações sobre consórcios de oleaginosas, contendo vários registros de experimentos em diversas situações. Esse conhecimento será usado como suporte para melhoria das pesquisas realizadas pela nossa equipe, além da possibilidade de aplicação das informações para a escolha da melhor cultura para cada ambiente específico, melhorando a produção, mesmo quando houver limitações de áreas agrícolas.

Um fator que evidenciou dificuldades na área de processamento de imagens utilizando a linguagem Java, foram as bibliotecas pesquisadas e utilizadas, pois não utilizam métodos diretos de manipulação dos objetos nativos da linguagem *java.awt.Image* e *java.awt.BufferedImage* e criam seus próprios objetos de manipulação de imagem e a conversão entre estes tipos não estava presente ou não é feita de forma intuitiva.

Enfim, a construção desse software permitiu um melhor gerenciamento dos experimentos de consórcio de plantas oleaginosas.

4. Conclusão

Os estudos para determinar quais cultivos consorciar necessitam de especialização, por exemplo, um centro de pesquisa implicando algumas vezes em um custo não muito acessível ao pequeno produtor.

Como possível alternativa a este problema, foi desenvolvido um software que analisa uma das variáveis fundamentais para determinação de qual ou quais cultivos podem ser praticados em uma determinada área, a radiação solar, fator de suma importância quando se trata de cultivos consorciados, por meio de processamento digital de imagens.

Buscando a customização do software de acordo com as necessidades, o software foi modularizado para permitir que profissionais da área ofereçam consultoria utilizando deste software como apoio na tomada de decisão e permitir que, através de demanda, outros serviços sejam acoplados (desenvolvidos) para incrementar esta oferta.

Assim sendo aliou-se a necessidade do pequeno produtor, a oportunidade de incentivos para estudos com biocombustíveis e tecnologia computacional viável de forma a contribuir com o desenvolvimento sustentável.

Referências Bibliográficas

- [1] D. Azevedo, E. F. Lima, J. W. Santos, F. A. S. Batista, L. Nóbrega, D. J. Vieira, and P. J. R. População de plantas no consórcio mamoneira/caupi. i. produção e componentes da produção. *Revista Oleaginosas e Fibrosas*, 3:13–20, 1999.
- [2] J. C. Ceballos and M. J. Bottino. Estimativa de radiação solar por satélite desempenho do modelo operacional g11.2. In *Anais do XIII Congresso Brasileiro de Meteorologia*. XIII Congresso Brasileiro de Meteorologia, 2004.
- [3] P. de Notícias G1. Lula critica ue por culpar brasil por alta do preço dos alimentos. Internet, acessado em 20 de maio de 2008, Maio 2008.
- [4] E. J. de Sá Parente. Processo de produção de combustíveis a partir de frutos ou sementes oleaginosas – biodiesel. Patente: Privilégio de Inovação, PI8007957, 14 de junho de 1983 (depósito); 08 de outubro de 1983 (concessão)., Outubro 1983.
- [5] E. R. Dougherty and R. A. Lotufo. *Hands-On Morphological Image Processing*, volume TT59. SPIE-International Society for Optical Engine, Jul 2003.
- [6] A. Freire. Mistura maior de biodiesel será obrigatória em julho, diz anp. Internet, acessado em 20 de maio de 2008, Março 2008.
- [7] R. C. Gonzalez and R. E. Woods. *Processamento Digital de Imagens*. Addison-Wesley Publishing Company, INC, 2 edition, 1992.
- [8] P. C. Neto, A. C. Fraga, J. B. Ferreira, H. P. de Castro, R. C. Avelar, D. C. Dourado, M. A. D. Júnior, S. Danfá, and A. A. Quintiliano. Mamona em consórcio com lavoura de café recepado. In *Anais do Iº Congresso Brasileiro de Plantas Oleaginosas, Óleos Vegetais e Biodiesel*. 2004.
- [9] C. M. Onyango and J. A. Marchant. Segmentation of row crop plants from weeds using colour and morphology. *Computer and Electronics in Agriculture*, 39:141–155, 2003.
- [10] P. Zanetti, S. Delfine, and A. Alvino. A mathematical approach for estimating light absorption by a crop from continuous radiation measurements and restricted absorption data. *Computers and Electronics in Agriculture*, 22:71–81, 1999.

Inspeção Visual de Placas de Circuito Integrado com Alta Densidade de Microcomponentes

Felipe Gomes de Oliveira (CNPq) e José Luiz de Souza Pio (Orientador).

Departamento de Ciência da Computação/ICE – Universidade Federal do Amazonas

fgo@dcc.ufam.edu.br – josepio@dcc.ufam.edu.br

Resumo

Este trabalho apresenta uma abordagem de inspeção visual para detectar ausência/presença de microcomponentes de superfície (SMC) em placas de circuito impresso (PCB). Propõe-se uma metodologia baseada em Estatísticas Bayesianas para detectar componentes ausentes, com mais qualidade e precisão. Nesta abordagem considera-se a ocorrência intensa de ruídos obtidos pelo processo de captura de imagens diretamente da linha de produção das PCB's. Os resultados obtidos demonstram que a metodologia é robusta e apropriada para o sistema real de inspeção visual industrial. Todos os casos testados de componentes ausentes foram corretamente detectados

Palavras-Chave: Inspeção Industrial, Visão Computacional, Estatísticas Bayesianas.

1. Introdução

Na economia atual as indústrias visam à competitividade. Mas para alcançar tal competitividade é de grande importância que as indústrias possam assegurar o aumento da produtividade com a garantia de qualidade de seus produtos [3].

A principal maneira de se garantir competitividade com qualidade é se dispor de um processo de inspeção que garanta um controle de qualidade rigoroso, com um mínimo de erros sobre um máximo de produção. Melhorias no controle de qualidade têm sido alcançadas por meio da inspeção automática com uso das tecnologias de visão de máquina [8].

A automatização do processo de inspeção de placas de circuito impresso tem se mostrado eficiente ao controle de qualidade, uma vez que se reduz o tempo de inspeção e aumenta consideravelmente o volume inspecionado [8,9]. Nesse contexto, este trabalho aborda o problema de detecção da ausência (ou da presença) de microcomponentes de superfícies (SMC) em dispositivos que utilizam a tecnologia SMD, como as placas mãe de computadores pessoais.

O objetivo principal deste trabalho é o desenvolvimento de uma abordagem para a identificação de microcomponentes, garantindo assim a velocidade, precisão e flexibilidade do controle de qualidade com competitividade dos produtos da empresa.

Este texto está organizado em seis seções. A seção seguinte descreve os sistemas de inspeção visual automática e suas aplicações no contexto deste trabalho. A Seção 3 oferece uma visão geral do estado da arte por meio dos principais trabalhos relacionados com a inspeção de componentes eletrônicos. A metodologia desenvolvida é mostrada na Seção 4. A seção 5 mostra os resultados experimentais e, por fim, na Seção 6 são apresentadas as conclusões deste trabalho.

2. Inspeção Visual Automática em Placas de Circuito Integrado

A inspeção visual automática utiliza sensores ópticos (câmeras) conectados a um computador para adquirir automaticamente uma imagem, armazená-la, e então executar algum tipo de processamento sobre ela, a fim de fazer uma avaliação em relação a um critério pré-definido [6].

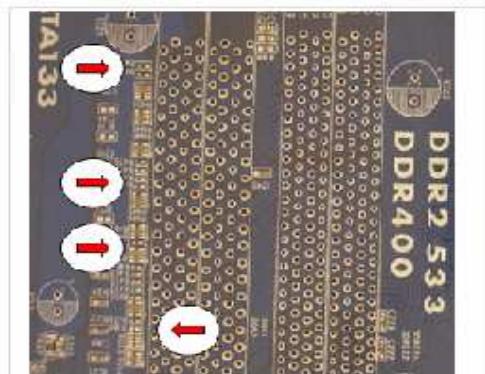


Figura 1 – Segmento de Placa apresentando ausência de componentes.

As setas indicam a posição de componentes ausentes na placa.

Neste trabalho é desenvolvida uma abordagem para a detecção da ausência/presença de microcomponentes (SMC) em placas de circuito impresso. A Figura 1 mostra um segmento de uma placa mãe de computador com seus componentes e a ausência de alguns outros, indicados pelas setas.

Esses microcomponentes são extremamente difíceis de serem observados pelo inspetor humano, sendo uma das principais causas de falhas nas placas ao saírem dos fornos de inserção das modernas linhas de produção.

A literatura científica é muito rica em exemplos de sistemas de inspeção visual para placas de circuito impresso [4,7,10]. A maior parte desses trabalhos tratam problemas relativos a identificação de trilhas de soldas ou a identificação de componentes mal posicionados. A seção seguinte destaca os trabalhos relacionados com a identificação da ausência de componentes e oferece uma visão geral do estado da arte.

3. Trabalhos Relacionados

Um dos primeiros trabalhos com trilhas de solda em placas de circuito impresso, foi o desenvolvido por Borba e Facon [2]. Os autores desenvolveram um método de inspeção que não utiliza o conhecimento prévio da placa e não utiliza um padrão a ser seguido como referência, detectando assim a falta ou excesso de cobre. Os autores acreditam que a imagem em escala de cinza não é a melhor solução, por isso utilizaram a imagem binarizada.

O sistema desenvolvido por Acciani e Brunetti [1] baseia-se em um Sistema de Inspeção Visual que utiliza rede neural com a finalidade de detectar defeitos encontrados nos terminais de solda nas placas de circuito integrado. As imagens das placas de testes são capturadas e processadas para extrair a região de interesse do diagnóstico. Três tipos de vetores de características são avaliados em cada região que são a exploração das propriedades da onda, as características geométricas e o pré-processamento das imagens.

O trabalho de Du e Dickerson [4] apresenta um sistema de inspeção automática para componentes passivos, para localizar componentes, medir seu tamanho e suas propriedades, suas bordas e detectar defeitos na superfície em ambos os lados. Os autores utilizam um algoritmo para detectar as bordas e os cantos. Por meio da segmentação da imagem, extrai-se e calculam-se as características do componente, como largura, tamanho, localização, orientação, etc., e com estas informações identifica defeitos, como componentes deslocados, pontos de ruptura na solda, mau contato na superfície. Essa técnica de inspeção não aborda informações quanto à ausência ou presença de componentes, pois na abordagem que é feita, cita a localização do

componente como uma das características de identificação de defeitos.

4. Metodologia

A Metodologia utilizada neste trabalho é decomposta em três etapas principais, que podem ser observadas na Figura 2. As etapas são:

- 1.Aquisição das Imagens.
- 2.Pré-Processamento das Imagens.
- 3.Classificação Probabilística da presença/ausência de componentes.

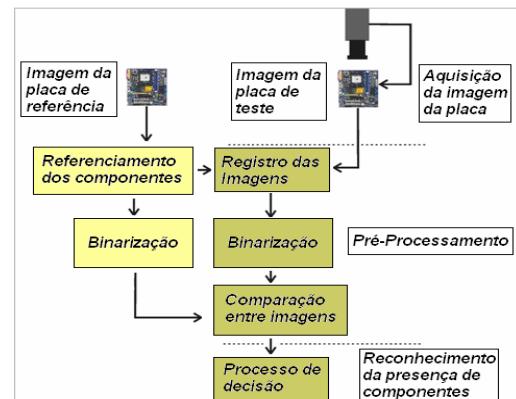


Figura 2 – Esquematização da metodologia adotada.

4.1. Aquisição das Imagens

Esse processo consiste na captura de uma imagem de uma placa por meio de uma câmera digital e na transferência das mesmas para um computador. Esse processo será realizado na aquisição da imagem referência e das imagens de teste.

4.2. Pré-Processamento das Imagens

Esta etapa pode ser subdividida em procedimentos menores que são:

- Conversão das Imagens: As imagens coloridas capturadas contidas no computador são convertidas para escala de cinza.
- Alinhamento das Imagens: O alinhamento de imagens pode ser definido como sendo um mapeamento entre as duas imagens de forma que essas diferenças sejam minimizadas [10]. As imagens da placa de referência e da placa teste são alinhadas, a fim de solucionar um problema muito comum nas esteiras, que são as mudanças súbitas que ocorrem no processo de captação.
- Binarização das Imagens: O processo de binarização é utilizado, pois segundo [6], o meio mais óbvio de distinguir objetos do fundo é por meio da seleção de um limiar que separe os dois grupos.
- Casamento de Padrões: Neste procedimento são feitas comparações entre os componentes das placas

de referência, de teste e a máscara binária. A máscara binária consiste em uma imagem de um componente ausente com área hipoteticamente perfeita, como pode ser observado na Figura 3.



Figura 3 – Máscara binária de um componente ausente [5].

As comparações entre os componentes e a máscara binária ocorrem por meio de uma operação lógica *and*, podendo ser observado na Figura 4.

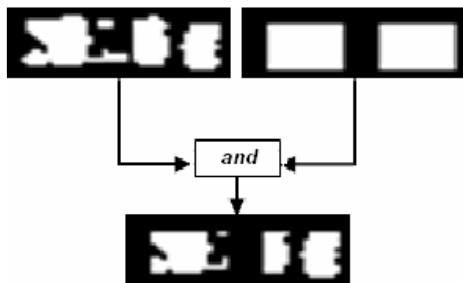


Figura 4 – Operação lógica *and* entre um componente e a máscara binária [5].

4.3. Classificação Probabilística da presença /ausência de componentes.

O processo de decisão do algoritmo baseia-se em um método estatístico de aprendizagem bastante consolidado na literatura, a Aprendizagem *Bayesiana*. A Aprendizagem *Bayesiana* calcula a probabilidade de cada hipótese, considerando os dados, e faz previsões de acordo com ela. Isto é, as previsões são feitas com o uso de todas as hipóteses, ponderadas por suas probabilidades, em vez de utilizar apenas uma única “melhor” hipótese. Desse modo, a aprendizagem é reduzida à inferência probabilística. Seja \mathbf{X} a representação de todos os dados, com valor observado \mathbf{x} ; então, a probabilidade de cada hipótese é obtida pelo teorema de *Bayes*, expresso abaixo:

$$P(W_i | \mathbf{x}) = [P(\mathbf{x} | W_i) \cdot P(W_i)] / P(\mathbf{x}), \quad (1)$$

onde

$$P(\mathbf{x}) = \sum_{i=1}^2 P(\mathbf{x} | W_i) \cdot P(W_i). \quad (2)$$

Nessa abordagem do teorema de *Bayes* são consideradas as seguintes hipóteses:

- W_1 é a hipótese denominada componente (presença).
- W_2 é a hipótese denominada almofada (ausência).

Sendo $P(W_1)$ definida como a razão entre $D1/(D1+D2)$ e $P(W_2)$ definida como a razão entre $D2/(D1+D2)$, onde $D1$ é a diferença do número de *pixels* brancos do componente da placa referência pelo número de *pixels* brancos da máscara binária e $D2$ é a diferença do número de *pixels* brancos do componente da placa teste pelo número de *pixels* brancos da máscara binária. Estas são as probabilidade *a priori* das hipóteses W_1 e W_2 respectivamente.

$P(\mathbf{x})$ representa a freqüência com a qual encontramos determinada característica, onde considera-se \mathbf{x} como um vetor de características formado a partir do número de *pixels* brancos extraídos de um componente da imagem.

A probabilidade $P(W_1 | \mathbf{x})$ representa a probabilidade que o componente da placa referência satisfaça a hipótese W_1 dado a característica \mathbf{x} . A probabilidade $P(W_2 | \mathbf{x})$ representa a probabilidade que o componente da placa teste satisfaça a hipótese W_2 dado a característica \mathbf{x} [9]. Utilizando o teorema de *Bayes*, queremos determinar qual a probabilidade do componente está na placa de teste dado que se sabe *a priori* que o componente está na placa referência. Em outras palavras, deseja-se determinar $P(W_2 | \mathbf{x})$.

Agora supondo que se queira fazer uma previsão sobre uma quantidade desconhecida N de componentes. Então temos a seguinte representação matemática da aprendizagem *bayesiana*:

$$P(\mathbf{x}) = \sum_i P(N | W_i) \cdot P(W_i | \mathbf{x}), \quad (3)$$

Onde têm-se que, $P(W_i | \mathbf{x})$ representa o valor resultante do teorema de *Bayes* para cada componente e $P(N | W_i)$ consiste na representação do conhecimento adquirido pelo método de aprendizagem a cada componente inspecionado.

5. Resultados Experimentais

Os experimentos realizados visaram mostrar a aplicabilidade da abordagem para a detecção da ausência de componentes em placas mãe de computadores pessoais. Para isso foram testadas 25 placas com 300 componentes cada, com componentes ausentes em diversas posições da placa. Com o ajuste dos parâmetros de probabilidade, foi possível detectar todas as ausências com a realização automática do processo de alinhamento de imagens. Para avaliar a robustez do método foram adicionados ruídos *gaussianos* e “salt and pepper” à imagem, a fim de verificar a robustez do método quanto a problemas nos processos inerentes a captura das imagens. Na adição de ruído *gaussiano* foram feitos experimentos com a densidade de ruído iniciando de 0.01 até 0.05, onde pôde-se observar o bom desempenho do algoritmo, que reconheceu todas as ausências da placa.

Para a inserção de ruídos “salt and pepper” o método detectou todas as ausências iniciando de 0.01 até 0.30 como pode ser observado na Figura 5.

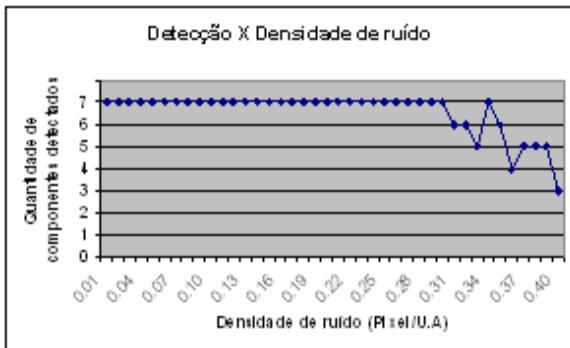


Figura 5 – Relação Detecção X Densidade de ruído.

Na Figura 6 pode ser observada a grande perturbação sofrida pela imagem da placa inspecionada na inserção de ruído *salt and pepper* com densidade 0.30 (Pixel/U.A), constatando-se a grande complexidade da abordagem em detectar com boa precisão a ausência/presença de microcomponentes em placas de circuito impresso.

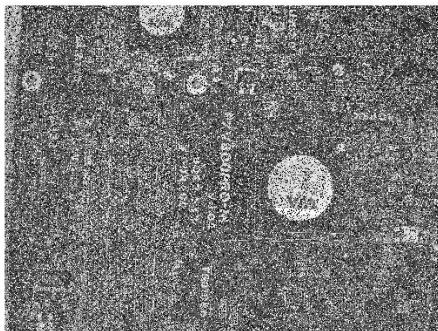


Figura 6 – Placa com densidade de 0.30 de ruído *salt and pepper*.

6. Conclusões

Este trabalho mostrou uma abordagem metodológica para a detecção de micro componente em SMD. Utilizou-se a estatística *Bayesiana* como ferramenta principal para a identificação de *pixels* que pertençam ao corpo de um componente ou que faça parte do ruído adicionado pelo processo ou dispositivo de captura da imagem. Um avanço importante no trabalho foi a realização automática do alinhamento de imagens, minimizando a interferência humana no processo de inspeção. Também destaca-se a identificação precisa dos parâmetros para a aplicação do teorema de *Bayes*, garantindo resultados mais precisos e confiáveis, mesmo com a inserção de ruídos gaussianos. A concretização do trabalho amplia consideravelmente as possibilidades de aplicação da

visão de máquina e da estatística *Bayesiana* para diversos outros tipos de inspeção industrial.

Agradecimentos

Os autores agradecem ao CNPq pelo apoio e incentivo dado ao projeto de pesquisa o qual este trabalho é vinculado.

Referências

- [1] ACCIANI, Giuseppe; Brunetti, Gioacchino e Fornarelli, Girolamo. Application of Neural Networks Optical Inspection and Classification of Solder Joints in surface Mount Technology, 2006.
- [2] BORBA, J.F. e Facon, J. A Printed Circuit Board Automated Inspection System, Anais do IEEE Midwest Symposium on Circuits and Systems, Rio de Janeiro, Brazil, pp. 69-72, Aug. 1996.
- [3] DA FONTOURA Costa, L. and Meriaudeau, F. Special Issue on Applied Visual Inspection, julho, 2002.
- [4] DU, Winnce Y. e Dickerson, Stephen L. Passive Component Inspection Using Machine Vision, 1998.
- [5] GONÇALVES de Araújo, Wilson. Detecção Automática de Componentes de Montagem em Superfície sobre Placas de Circuito Impresso. 2008. Dissertação (Mestrado em Engenharia Elétrica) – Curso de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Amazonas, Manaus.
- [6] GONZALEZ, Rafael C. e Woods, Richard E. Processamento de Imagens Digitais. Rio de Janeiro. EDITORA EDGARD BLUCHER, 2000.
- [7] HATA, S. Practical Visual Inspection Techniques: Optics, Micro-electronics and Advanced Software Technology. Anais da International Conference on Pattern Recognition 2000, vol. IV, páginas 114-117.
- [8] SASAI, M. Inspection Technique for Flexible Manufacture. Electronic Packaging Technology, vol. 2, páginas 41-46, 1998.
- [9] MARINOVO Doro, Marcos. Sistemática para Implantação da Garantia da Qualidade em Empresas Montadoras de Placas de Circuito Impresso. 2004. Dissertação de Mestrado – Programa de Pós-Graduação em Metrologia Científica e Industrial, Universidade Federal de Santa Catarina, Florianópolis.
- [10] MARTINEZ ,Valguima V. V. A. Odakura_; Campos, Geraldo Lino. Uma Técnica para Alinhamento de Imagens de Documentos Antigos, 2006.

VHuP: a Tool to Visualize Virtual Humans

Diogo Strube de Lima, Henry Braun, Soraia Raupp Musse

*Computing Science Department, Postgraduate course in Computer Science
PUCRS, Av. Ipiranga, 6681, building 32, 90619-900 Porto Alegre, RS, Brazil
{diogo.lima, henry.braun}@cpph.com.br soraia.musse@pucrs.br*

Abstract

This paper presents an application which aims to easily provide visualization and post processing of virtual human simulations. The interactive interface reproduces, in a real-time frame rate (30 frames per second, FPS), a previously calculated scenario of characters simulation, including multiples characters and their trajectories, on different environments. Features to control the light sources, shadows and weather effects provide a wide range of possibilities for visualizing and video recording the simulation data. The application also facilitates the generation of analyses data, generating information for a future application, as for instance ground truth.

1. Introduction

Crowd simulation is an area of computer graphics and artificial intelligence concerned with the process of simulating a large number of models. Nowadays, virtual human crowd's simulation is an important topic of research since it is crucial for some applications as for instance, security, education and entertainment. Such simulations, as the number of 3D objects involved can be huge, are not trivial to render at high resolution on standard computers and graphic cards [1].

Current methods to simulate virtual humans and crowds are often mediated by local rules [2], forces [3], flows [4] or hybrid approaches [5]. Usually, these methods consider the characters as simplified geometries such as points, spheres or cubes. This abstraction is important to facilitate the usage of complex mathematical models, however it can create visualization artifacts when those simplified geometries are replaced by high polygon characters, that perform a variety of animations.

Current approaches [1, 5, 6] for crowd simulations are designed for a specific use, and normally the simulations prototype also includes the visualization

features, limiting the integration of different algorithms. In our lab (Virtual Humans Laboratory, at PPGCC, PUCRS), we used an approach which separates the visualization from simulation aspects. In this case, the advantage was the possibility of integrating several simulators of virtual humans, by mixing their output in a same visualized scenario. Our main idea is to create an independent application for the real-time visualization of virtual human simulations. That application was named Virtual Human Player (VHuP).

The paper is organized as follows: related works are resumed on the next section, followed by section 3, which has an overview of the VHuP architecture and its key components. Section 4 presents some results obtained developing and testing the application. On section 5 the ideas for future work and new components are described.

2. Related Work

Several researches on virtual human visualization have been provided in last years. Hamill and O'Sullivan [1] resumes a number of methods available for culling such as portal, occlusion horizons, occlusion shadow volumes and the simple bounding box tests.

The vast type of environment, such indoors and outdoors, used on the crowd simulation makes common the use of multiple techniques combined. The Virtual Dublin [1] uses culling and collision techniques combined with the trade of model details for texture details, reducing the building's polygons, but increasing the memory need.

The approach of high-level wayfinding using agent's communication and roles, from Pelechano and Badler [5], proposes an architecture that combines and integrates MACES and PMFserv frameworks increasing the crowd behavior accuracy.

The framework created by Pettré, Ciechomski, Maïm, Yersin, Laumond and Thalmann [6] achieved

the real-time visualization of crowd simulations combining levels of detail (LOD) techniques and billboards, dramatically reducing the cost of animations updates and deformations on the models.

Another work from Thalmann and Pablo [7] presents a rendering engine to visualize a crowd of virtual humans in real-time. The virtual humans are sorted by their distance to the observer into render fidelities groups. The engine combines impostors and different rendering acceleration techniques, such as caching schemas, levels of detail, shader and state sorting.

3. VHuP Architecture

VHuP performs the real-time visualization of virtual humans and crowd, using a group of open sources platforms and toolkits. This group is coordinated by a core, which has the main functionalities of the application. All the services are built over this core, providing flexibility and an easy update capability.

The features are implemented maintaining the primary goal of the application: the ability to visualize a variety of crowd and virtual human simulations algorithms used on our research lab. The idea is making the visualization process simple, reducing the time of developing specific interfaces and avoiding any rework. The **Figure 1** is an overview of the VHuP architecture and is followed by an explanation of each component.

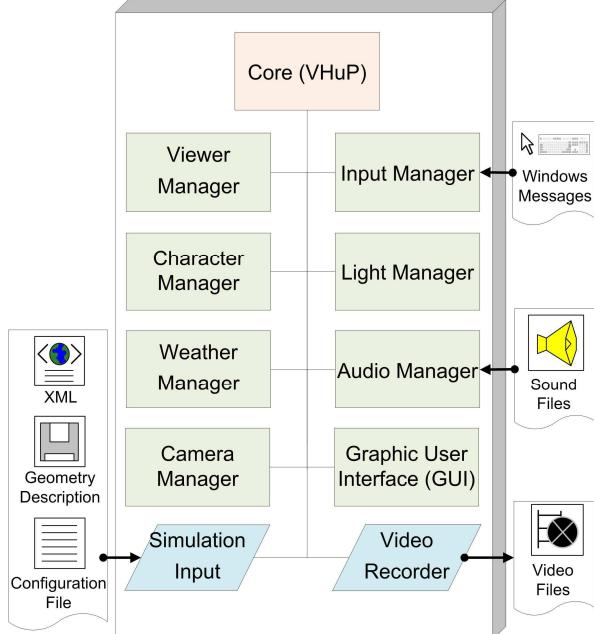


Figure 1. VHuP architecture overview.

The **Core (VHuP)** is the main component of the application, responsible for executing and managing the other components. The core coordinates the application logic: updates, draws and events. This is performed using the Open Scene Graph (OSG) [8] graphic toolkit as rendering engine. It also contains several important functionalities and interfaces for communication between the components and OSG.

Our application uses a composite view model, in other words, the application is allowed to place up to five views in the scene, including a mini map view and a record view. Those views, controlled by the **View Manager**, provide a greater perspective of the simulation allowing the possibility to focus on areas of interest. This component handle the window resize events, cameras and every scene data attached to each view.

The characters are structured in three parts: model, bones and animations. The **Character Manager** organizes this structure in memory and is in charge of all actions related to characters along the simulation. The component uses the 3D Character Animation Library (Cal3D) [9] through the Cal3D adapter for OSG (osgCal2) [10] to perform animations. It also provides hardware skinning, avoid streaming dynamic data to GPU, and animation blending, smoothly playing animations.

The **Light Manager** allows determining how the environment's illumination should be. It is possible to add and modify lights sources customizing the scene and increasing the simulation realism.

The shadows created by the light sources are also controlled by this component. These shadows can be modified, changing the shadow appearance, type, quality or even disabling it to obtain the desired tradeoff between performance and graphic quality.

In order to increase the graphic quality and realism of the visualization, there is also the **Weather Manager**. This component is capable of creating weather effects such as fog, rain, snow and clear weather.

The **Camera Manager** component allows navigating in the scenario easily. It uses the OSG camera architecture making it possible to set camera manipulator to each point of view in the application. Our default camera manipulator is similar to strategy games cameras.

The Microsoft Windows-based application uses messages in order to handle events. Those messages, sent by other applications, keyboard or mouse, are handled by the **Input Manager**. The features of this component are the input buffer, computer pens and parallel communications. The input buffer supports

multiple pressed keys. Computer pens can be used and the pressured sensitivity is also controlled. The parallel communication makes possible an interaction with other applications such as eyes and gesture recognition.

Audio Manager performs the simulation audio and uses the Ambiera IrrKlang Audio Library [11] to provide support to different audio file formats such as MP3, OGG and WAV. The additional files are previously loaded before the simulation rendering avoiding possible slowdowns.

VHuP has a friendly **Graphic User Interface (GUI)** granting access to the application functionalities easily and in a fast way. The **GUI** architecture is component based and was built over the OSG event handlers classes. This approach allows compatibility with OSG applications and new components are added with no effort. Each component of the **GUI** have a set of events related to it, the actions to each event are easily customized. The mouse click is an event example and coloring a button when it's clicked is a simple action. The architecture divides the **GUI** components on three layers: window, forms and objects. Buttons, images and textboxes are examples of objects. A group of objects are organized in forms, and the **GUI** scope is the window.

The **Video Recorder** uses a separated view to record on a set of image files. Those images are recorded on a chosen frame rate and with a custom graphic quality, relative to the recording view. The files are compressed and filtered using any video software, as the Virtual Dub, resulting on a video file.

This **Simulation Input** receives all the simulation data files and is responsible to read and parse the data. These files are the xml defining the simulation, the application configuration xml and the meshes. Those meshes are supported on different formats, such as OSG, 3DS and OBJ.

4. Results

The results were obtained executing VHuD on a AMD 64 x2 Dual Core 4200+, 2GB RAM DDR, equipped with a GeForce 8800GTS OC. The frame-rate difference (measured in FPS) between the higher and lower graphic quality is easily visible when a simulation with a small number of characters is rendered.

When the number of agents increases, the computational time for high quality rendering is decreased. This shows that using finer techniques, shadows and weather effects are not the bottleneck of the application. **Figure 2** shows a graphic where FPS

obtained as a function of changing rendered number of characters is presented.

Frames Per Second

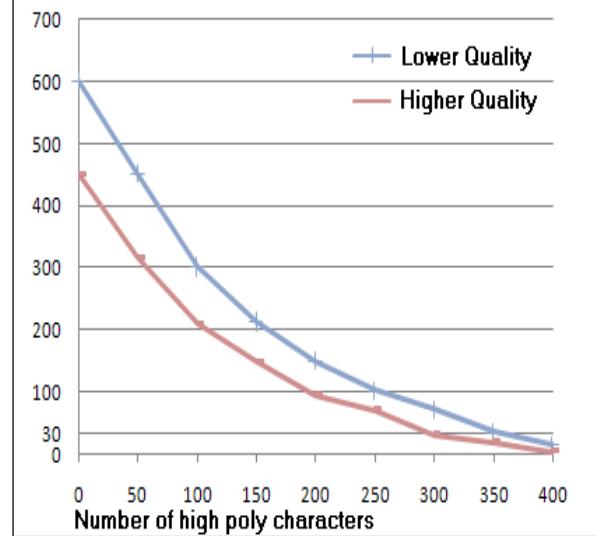


Figure 2. FPS related to the number of characters.

The application's bottleneck is visible when rendering more than 300 characters. The virtual humans animation's update is bounded to the computer processing unit (CPU), requiring a large amount of process. This issue makes the higher quality, illustrated at **Figure 3**, with the best tradeoff between performance and graphic quality.



Figure 3. VHuD at high quality.

5. Future Work

The obtained results were acquired with the first VHUP version. On the second version new techniques will be studied and implemented to improve the graphic quality and to increase the number of characters rendered in real-time.

The graphic quality could be improved using shaders to better render lights and different materials, so rendering realistic water and fire effects becomes possible. Different techniques for creating shadows and weather effects could be performed. Higher polygon characters could also be used, adding LOD and culling techniques [1].

Those techniques, combined with impostors and rendering acceleration techniques [7], could also increase the number of character rendered in real-time. At last, creating a database for characters animations and models, could increase both the graphic quality and the performance of the application.

6. Acknowledgements

We thank all the support from Marcelo Paravisi and Rafael Rodrigues during the development and test of the VHUP. Besides being authors of the VHSimul project, they created unique simulations and gave important ideas for the application. This work was developed in collaboration with HP Brazil R&D.

7. References

[1] J. Hamill, C. O'Sullivan, J “Virtual Dublin – A Framework for Real-Time Urban Simulation”, [WSCG, 2003](#).

[2] S. R. Musse, and D. Thalmann, “Hierarchical model for real time simulation of virtual human crowds”, *IEEE Transaction on Visualization and Computer Graphics*, 2001, pp. 152-164.

[3] D. Helbing, I. Farkas, and T. Vicsek, “Simulating dynamical features of escape panic”, *Nature*, 2000, pp. 487-490.

[4] A. Treuille, A. Lewis, and Z. Popovic, “[Model reduction for real-time fluids](#)”, SIGGRAPH, 2006.

[5] N. Pelechano, and N. Badler, “Modeling Crowd and Trained Leader Behavior during Building Evacuation”, *IEEE Computer Graphics and Applications*, Volume 26, 2006, pp. 80-86.

[6] J. Pettre, P. H. Ciechomski, J. Maïm, B. Yersin, J. P. Laumond, and D. Thalmann, “Real-time navigating crowds:

scalable simulation and rendering”, *Computer Animation and Virtual Worlds*, Volume 17, 2006, pp. 445–455.

[7] P. S. H. Ciechomski, and D. Thalmann, “Rendering Massive Real-Time Crowds”, *Thèse n° 3534 - École Polytechnique Fédérale de Lausanne*, 2006.

[8] Open Scene Graph (OSG), *open source high performance 3D graphics toolkit*, <http://www.openscenegraph.org/>, accessed July 20, 2008.

[9] 3D Character Animation Library (Cal3D), *skeletal based character animation library*, <https://gna.org/projects/cal3d/>, accessed July 20, 2008.

[10] Cal3D adapter for OpenSceneGraph (osgCal2), *adapter of cal3d for use inside OSG*, <http://osgcal.sourceforge.net/>, accessed July 20, 2008.

[11] Ambiera IrrKlang Audio Library, *high level 2D and 3D cross platform sound engine and audio library*, <http://www.ambiera.com/irrklang/>, accessed July 20, 2008.

Desenvolvimento de um Programa Computacional para Medição do Tamanho do Grão de Materiais Metálicos.

Francisco Roger L. CAETANO; Magnaura M. dos SANTOS
José Pinheiro QUEIROZ-NETO (Orientador).

*rogerl.caetano@gmail.com, magnaurasantos@gmail.com
pinheiro@cefetam.edu.br*

Abstract

Nos metais, parte das suas propriedades macroscópicas tem forte ligação com sua microestrutura e para relacioná-los utiliza-se a metelografia quantitativa, que consisti em técnicas de análise microestrutural da constituição do material metálico. A precisão das análises metalográficas de materiais está relacionada à medição correta dos grãos, que são pequenos cristais que compõem a microestrutura do material. Contudo, a quantificação manual da medição dos grãos é uma tarefa lenta, trabalhosa e sujeita a falha humana. Com o auxílio de técnicas de visão computacional é possível automatizar o processo de maneira a torná-lo mais rápido, preciso e confiável. Portanto, este trabalho visa desenvolver um programa computacional para a medição automática do tamanho do grão de materiais metálicos.

1. Introdução

O Brasil tem tido um grande avanço nas pesquisas de materiais com forte crescimento nos últimos anos. Com a evolução dos materiais, atualmente convivemos com os Monitores de Plasma, Monitores de Cristal Líquido - LCD, celulares de alto desempenho e novos insumos e equipamentos industriais [1].

A Metalografia é um conjunto de métodos e técnicas utilizados na análise de materiais, proporcionando identificar um dado tipo de material a um custo relativamente baixo. Basicamente este ensaio consiste em: corte da amostra, embutimento, lixamento, polimento, ataque químico e a observação microscópica que revela as interfaces entre os diferentes cristais [2]. Esta observação microscópica é desenvolvida, em geral, de maneira não automática, levando o pesquisador à utilização de um tempo considerável para identificar um determinado material.

A microestrutura de um material é constituída principalmente dos tamanhos de grão que a compõe, e tais medições são realizadas com muita freqüência

tanto nas indústrias como nas instituições de pesquisa. Contudo, a quantificação manual da medida do grão é uma tarefa lenta, trabalhosa e sujeita a falhas e fadiga humana.

Alguns pesquisadores têm utilizado a Visão Computacional para automatizar o processo de medição do grão. Pode-se observar em [3], que técnicas de visão computacional são utilizadas para propor uma metodologia semi-automática de análise e medição do tamanho do grão, através de análise em imagens obtidas em um sistema microscópio/câmera, e processadas de tal maneira a permitir uma visualização segmentada das características que se deseja fazer a medição.

Nos trabalhos de Fazano [4,5], um sistema é utilizado para medição automática do grão, porém o autor não se preocupou em explicitar maiores detalhes quanto às técnicas utilizadas no processamento da imagem e no método de contagem, limitando-se a informar que o sistema efetua as medições conforme a norma ASTM-E112 [6]. O método proposto neste trabalho se baseia em [5], utilizando técnicas mais simples para efetuar a medição.

Este trabalho contribui para o processo automático de medição de grão de materiais metálicos, utilizando um sistema que inclui uma câmera acoplada a um microscópio para aquisição da imagem.

2. Grãos e Microestrutura

Materiais são substâncias com propriedades que as tornam úteis na construção de máquinas, estruturas, dispositivos e produtos. A microestrutura está profundamente ligada com as propriedades dos materiais, e delas dependem a aplicação e a utilização de cada equipamento desenvolvido com um ou mais tipos de materiais.

Os materiais metálicos são cristalinos, isto é, têm suas ligações atômicas organizadas em estrutura cristalina formando arranjos que assumem uma configuração geométrica que, dependendo do elemento, pode formar diversas geometrias. Nos

metais, a configuração mais comum é a cúbica de face centrada (CFC) ou a hexagonal centrada (HC), respectivamente as Figuras 1(a) e 1(b).

Dos diversos tipos de defeitos na estrutura cristalina destacam-se os grãos, que são pequenos cristais poliédricos com diferentes orientações cristalográficas, isto é, numa rede cristalina, não alinhada, existem inúmeras orientações, onde cada uma constitui um cristal poliédrico ou grão. O conjunto de todas as imperfeições, em especial os grãos, forma a microestrutura, e a visualização desta é alcançada através da metalografia.

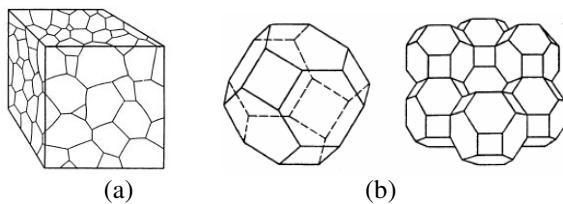


Figura 1 - Estruturas cristalinas. Extraída de [2].

3. Metalografia

A Metalografia é a ciência que estuda a morfologia dos metais, e nesta se encontra o processo de preparação da amostra do material que pode ser resumido em etapas. A primeira etapa é o Corte, que é a escolha e localização da seção a ser estudada que ficará a critério do analista que será guiado pela forma e dados que se quer obter e outras considerações da peça em estudo [6]. O Embutimento é a próxima fase e pode ser a quente ou a frio dependendo do ponto de fusão e dureza do material que se irá trabalhar. Em seguida, o Lixamento é uma etapa em que se retiram as impurezas mais grosseiras do material como defeitos de corte e defeito de fabricação. No Polimento, é a etapa em que se eliminam os riscos da ultima lixa e deixa a peça espelhada e pronta para a próxima etapa, a do ataque químico, que consiste no contato do material com o ácido que reage com o elemento que envolve todos os grãos, formando profundos sulcos ou depressões delineando os grãos e revelando-os pelo seu contorno, deixando a amostra apta para aquisição via sistema microscópio/câmera.

4. Microscopia Óptica

A Microscopia é a fase onde se visualiza, identifica e analisa as microestruturas dos metais qualitativamente, isto é, por suas características visuais. Há três tipos de microscopia que são utilizados em grande extensão na metalografia: a microscopia óptica (MO), a microscopia eletrônica de varredura

(MEV) e a microscopia eletrônica de transmissão (MET), e desses tipos de microscopia se destaca o microscópio óptico metalográfico por sua simplicidade e custo em comparação com os demais. Para analisar especificamente os metais utiliza-se o microscópio óptico de reflexão por sua baixa interação com o metal. Durante a análise é de suma importância possuir condições para uma boa visualização como posicionamento, fixação e firmeza do microscópio, a amostra deve estar plana e bem focada. Todo este processo é para reduzir os ruídos na imagem e aumentar a definição dos grãos garantindo uma boa precisão na análise.

5. Método de Medição dos Grãos

A microestrutura tem grande influência nas propriedades macroscópicas do material, e para relacioná-las quantitativamente, isto é, através de números, utilizando a metalografia quantitativa, que consiste em técnicas de análise microestrutural fundamentadas na estatística, aplicada a uma imagem do microscópio ou micrografia de uma amostra plana ou transversal do material, com o objetivo de representar numericamente as características da micrografia com as propriedades. O processo de quantificação de cada característica da microestrutura é padronizado pela ASTM (*American Society Testing and Material*) [6].

A norma técnica ASTM E-112 padroniza a quantificação do tamanho médio do grão, que consiste em traçar três círculos concêntricos sobre a micrografia ampliada em 100 vezes seguindo a condição de que deve haver no mínimo 50 grãos em cada círculo (Figura 2).

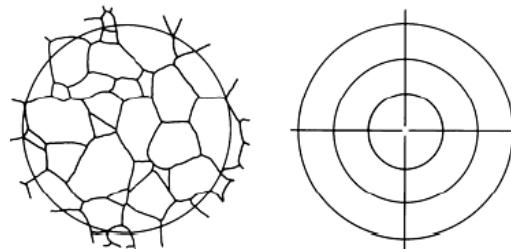


Figura 2 – Uso da norma ASTME112. Obtida de [22].

Desta forma, calcula-se a razão da quantidade de grãos contidos no círculo pela área do círculo, de cada círculo em polegadas quadradas, e em seguida é calculado a média dos resultados. O valor obtido representa o número de objetos da microestrutura por unidade de área, dado por N_a que é utilizado na Equação 1:

$$N = (\log N_a / \log_2) + 1 \quad (\text{Equação 1})$$

Onde N é o número da ASTM que, através de uma tabela e figura padrão, indica um valor percentual entre 0 e 10 que representa uma micrografia com grãos semelhantes para comparação a um valor de referência em micrometros e determinação do tamanho do grão. Entretanto, para alcançar a condição da norma que estabelece um ou dois círculos para ultrapassarem a fronteira da imagem deve-se apenas utilizar os círculos dentro da imagem. Porém, se o diâmetro do primeiro círculo exceder o tamanho da imagem, então deve-se utilizar o Método de Hilliard [5], que consiste em traçar três círculos concêntricos, como na Figura 2, de comprimento conhecido, em centímetros, sobre a micrografia de qualquer ampliação e contar o número de fronteiras de grãos interceptados pela linha para substituir na Equação 2. Contudo, é importante que obedeça à condição de que deve haver mais de seis interceptações do círculo com os grãos.

$$D = \text{Comprim. Circ.} / (\text{Conta.x Ampl.}) \quad (\text{Equação 2})$$

O resultado da Equação 2 é substituído na Equação 3:

$$G = -10,00 + 6,64 \log(D) \quad (\text{Equação 3})$$

Onde G é o número da ASTM em micrometros do tamanho médio do grão pelo Método de Hilliard.

Apesar de simples, a quantificação manual é trabalhosa, porque é necessário quantificar diferentes áreas da micrografia e diferentes micrografias do mesmo material obedecendo às restrições para um bom nível de confiança e precisão dos resultados através de uma média. Também na análise está sujeita a falha humana, pois, com o avanço contínuo da repetição do processo, cresce os erros causados por fadiga.

6. Implementação e Resultados

Utilizando o que recomenda a Norma da ASTM E-112, o processo de automatização da quantização inicia a partir da digitalização da micrografia através de um sistema de câmera acoplada a um microscópio óptico metalográfico e conectado ao computador, no Laboratório de Materiais e Componentes Eletrônicos do CEFET-AM. Para o inicio do sistema é de suma importância que a micrografia tenha excelente iluminação, contraste, foco e, sobretudo, nitidez. Após selecionar a micrografia, a imagem é quantizada para tons monocromáticos, e em seguida suas linhas são

segmentadas através do algoritmo de CANNY EDGE, resultando numa imagem preta e branca, sendo preta nas bordas de objetos e branca em todo o resto da imagem (Figura 3). Até esta etapa o sistema apenas prepara a micrografia para análise.

O sistema obedece à condição de que deve haver no mínimo 50 grãos por círculo, exigida pela norma técnica, porém, os grãos não são uniformes e variam muito de material para material, então, o sistema ajusta o raio de acordo com o tamanho de grão através do algoritmo de ajuste, desenvolvido neste trabalho como um software livre em linguagem C++. O algoritmo baseia-se em duas medições de grãos, a primeira é a medição grosseira do grão que consiste na varredura linear de pixel a pixel contando o número de fronteiras de grãos encontradas (Figura 3(b)), utilizado para cálculo da razão entre o comprimento da linha e da contagem conforme a Equação 4. O método irá se repetir 10 vezes aleatoriamente, para obter um resultado médio preciso.

$$D = \text{Comprimento da linha} / \text{Contagem} \quad (\text{Equação 4})$$

A segunda é a medição simples do grão que consiste na varredura circular de pixel a pixel de três círculos concêntricos (Figura 3(c)) de cada vez, no qual o raio do primeiro círculo é o resultado da Equação 4 e os demais são 50% e 150% do primeiro círculo. Conta-se o número de contorno de grãos interceptados durante a varredura para utilizar na Equação 2, porém, considera-se a ampliação igual a 1. O processo repete-se para uma média dos resultados dos três círculos.

O resultado da Equação 2, com ampliação igual a 1, é o valor mínimo do raio para o ajuste, o qual consiste no aumento aritmético do raio na razão de 50% do raio mínimo até a condição verdadeira da restrição da norma ASTM. No entanto se o diâmetro do primeiro raio ultrapassar o tamanho da imagem o sistema adota o procedimento do Método de Hilliard.

Na condição de haver pelo menos 50 grãos no primeiro círculo, o sistema inicia a quantificação do grão calculando a razão entre o número de grãos e a área do círculo, de cada um dos três círculos concêntricos para um resultado médio. Este resultado é substituído por N_a na Equação 1. O valor de N_a acessa, na tabela ASTM, o valor do tamanho médio do grão e a micrografia correspondente. Como saída do sistema obtém-se a micrografia segmentada por bordas, o número da ASTM, o tamanho médio do grão e a micrografia padrão, de acordo com a ASTM.

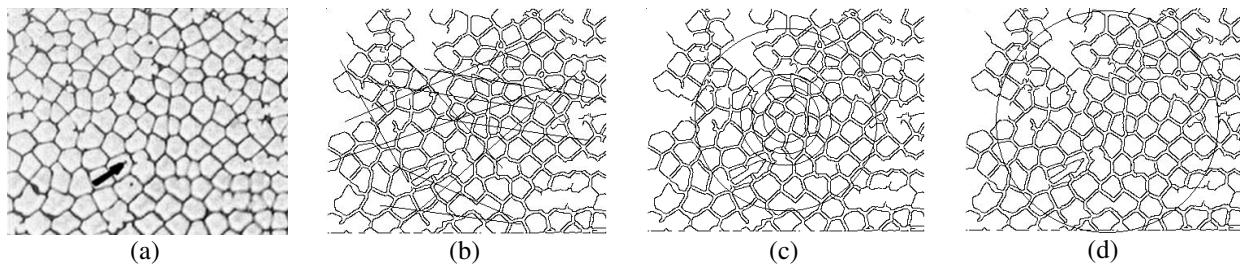


Figura 3 - Micrografia C - Amostra do material Pb-2Sb-8 obtida de [6]. Imagens resumindo o processo (a) Imagem Monocromática (b) 10 Linha Aleatórias para Analise Linear (c) Três Círculos Concêntricos Para Analise Circular (d) Contagem de 50 grãos para a Norma ASTM E112

Caso não haja menos 50 grãos no primeiro círculo, o programa aproveita o resultado da Equação 2 obtida através do processo de varredura circular, porém agora com o valor da ampliação da micrografia aplicada na formula, para substituí-la na Equação 3 e obter o tamanho médio do grão. Em micrografias com o número total de grãos inferior a 50, o sistema adota a medição de G (Equação 3).

Para o teste e comprovação do sistema foram efetuados experimentos com as Amostras A, B, C (Figura 3) e D extraídas de [6], sendo que esta última apresenta menos de 50 grãos no primeiro círculo para o teste da quantização pelo Método de Hilliard.

Os resultados dos cálculos de tamanho dos grãos são apresentados na Tabela 1. A medição manual (análise a olho nu na imagem) foi efetuada por um professor do CEFET-AM com 23 anos de experiência na área e confrontada com os valores obtidos pelo sistema desenvolvido neste trabalho.

A seguir o tamanho médio do grão, retirado de [6] e com alta precisão, foi comparado ao resultado do programa, observou-se que há uma pequena diferença, porém justificável, pois a precisão do programa é fundamentada na repetição do método em várias micrografias do mesmo material e não apenas em uma. Além disto, não é possível afirmar quanto ao grau de certeza da medição, uma vez que a quantidade de amostra foi pequena, e considerando que a medição manual não configura um valor exato, pois está sujeito a erros.

Amostras da Micrografia	Tamanho do Grão (Manual)	Tamanho do Grão (Automático)	Erro
A	22,05	26,278	19%
B	60,88	66,852	9%
C	108,33	122,47	13%
D	33,09	30,36	8%

Tabela 1 – Resultados comparando o método manual de medição e sistema automático proposto

5. Conclusão

A quantificação manual das medidas dos grãos é uma tarefa lenta e sujeita a falhas. Com o auxílio da visão computacional e métodos de processamento da imagem digital, este trabalho demonstrou que é possível automatizar o processo de medição dos grãos de maneira a torná-lo mais rápido e confiável. As técnicas utilizadas permitem a medição do tamanho do grão de maneira simples e eficaz, contudo este ainda é um resultado que precisa ser validado através de uma quantidade maior de amostras com mais experimentos e diferentes tipos de micrografias.

Agradecimento

Agradecemos a FAPEAM pelo auxílio através do Programa de Apoio a Iniciação Científica (PAIC).

Referências

- [1] REZENDE, Sérgio., *Materiais e dispositivos*, Ed. Livraria da Física, 2ed, ISBN 8588325276, 2004.
- [2] PADILHA, F. A e Ambrozio filho, F., *Técnicas de análises microestrutural*, Hemus livraria, distribuidora e editora 2004.
- [3] DIÓGENES, A. N., Hoff, E. A., Fernandes, C. P., “Grain size measurement by image analysis: An application in the ceramic and in the metallic industries”, Proceedings of 18th International Congress of Mechanical Engineering - COBEM, November 6-11, 2005, Ouro Preto, MG.
- [4] FAZANO, C. A. , *A Aplicação da Análise de imagem na quantificação das características microestrurais de materiais*, Revista Analytica•Outubro/Novembro 2005•Nº19.
- [5] FAZANO, C. A., “Determinando-se o tamanho de grãos em ligas de cobre-zinco através da análise de imagem”, Revista Analytica • Fevereiro/Março 2007 • N°27.
- [6] *ASM Metals HandBook - Metallography and Microstructures - Volume 09*, Ed. Metals Handbook 1985.